

Trabajo Fin de Grado:

Detección y reconocimiento de señales de tráfico



Autor: Daniel Cruzado Hernando

Tutor: Fernando García Fernández

Grado en Ingeniería Electrónica, Industrial y Automática

Universidad Carlos III de Madrid

Septiembre, 2015

ACRÓNIMOS	5
AGRADECIMIENTOS	6
1. INTRODUCCIÓN	7
2. PROBLEMÁTICA ACTUAL	8
2.1. CIFRAS EN ESPAÑA	9
2.2. CAUSAS	11
2.2.1. <i>La velocidad</i>	11
2.2.2. <i>El alcohol y las drogas</i>	12
2.2.3. <i>La distracción</i>	13
2.2.4. <i>Las infracciones</i>	13
3. ESTADO DEL ARTE	15
3.1. DETECCIÓN DE SUEÑO O FALTA DE ATENCIÓN AL VOLANTE	15
3.2. DETECCIÓN DE CAMBIO DE CARRIL INVOLUNTARIO	15
3.3. ADVERTENCIA DE ÁNGULO MUERTO EN EL ESPEJO RETROVISOR	16
3.4. APARCAMIENTO AUTOMÁTICO	17
3.5. RECONOCIMIENTO DE OBJETOS	17
3.6. CONTROL DE DESCENSO DE PENDIENTES	18
3.7. DETECCIÓN Y RECONOCIMIENTO DE SEÑALES DE TRÁFICO	19
4. DESCRIPCIÓN GENERAL DEL SISTEMA	21
4.1. INTRODUCCIÓN A LA APLICACIÓN	21
4.2. HARDWARE	22
4.2.1. <i>Cámara SJ4000</i>	22
Conexión de la cámara	23
4.2.2. <i>IVVI 2.0</i>	24
4.3. SOFTWARE	27
4.3.1. <i>QtCreator</i>	27
4.3.2. <i>OpenCV</i>	27
4.3.3. <i>ROS</i>	27
5. ESTRUCTURA DEL PROGRAMA	29
6. CALIBRACIÓN DE LA CÁMARA	31
7. DETECCIÓN DE SEÑALES	36
7.1. ESPACIO DE COLORES	36
7.2. SEGMENTACIÓN POR COLORES	37
7.3. POST-PROCESAMIENTO	41
7.4. DETECCIÓN DE CÍRCULOS Y TRIÁNGULOS	44
8. RECONOCIMIENTO DE SEÑALES	47
8.1. CLASIFICADOR DE CASCADA DE CARACTERÍSTICAS TIPO HAAR	47
8.2. PREPARACIÓN DE LAS MUESTRAS	48
8.2.1. <i>Muestras positivas</i>	48
8.2.2. <i>Muestras negativas</i>	49
8.3. ENTRENAMIENTO	50
8.4. RECONOCIMIENTO	53

9.	RESULTADOS	57
9.1.	TIEMPO DE EJECUCIÓN	57
9.2.	DETECCIÓN	58
9.3.	RECONOCIMIENTO	59
10.	PRESUPUESTO	63
11.	TRABAJOS FUTUROS	65
12.	CONCLUSIONES	67
13.	BIBLIOGRAFÍA	68

Acrónimos

- ADAS: Advanced Driver Assistance Systems
- AVAD: Años de Vida Ajustados por Discapacidad.
- CBM: Color Barycenter Model
- CCD: Charge-Coupled Device (Dispositivo de carga acoplada).
- GNSS: Global Navigation Satellite System (sistema global de navegación por satélite)
- HOG: Histogram of Oriented Gradients (histograma de gradientes orientados)
- IMU: Inertial Measurement Unit (unidad de medición inercial)
- IVVI: Intelligent Vehicle based on Visual Information (vehículo inteligente basado en información visual)
- OCR: Optical Character Recognition (reconocimiento óptico de caracteres)
- ROS: Robot Operating System
- RTSP: Real Time Streaming Protocol (protocolo de flujo en tiempo real)
- SVM: Support Vector Machine (máquina de soporte de vectores)
- OpenCV: Open source Computer Vision

Agradecimientos

A mi familia, sin la cual no habría llegado aquí,

A todos aquellos que me han llevado en coche para que grabara cada señal de Madrid,

A mi tutor Fernando García, quien ha sabido desbloquear el trabajo en sus malos momentos.

1. Introducción

En los últimos tiempos la electrónica y los sistemas de ayuda a la conducción están ganando cada vez más importancia. La adición a los vehículos de estos sistemas de ayuda no es ya por un simple tema de comodidad, sino por la reducción del alto número de accidentes que se producen a diario por el uso de los automóviles. Estos sistemas inteligentes están desarrollándose tanto en los últimos tiempos que ya se ha podido ver a coches conduciéndose por sí mismo en las mismas calles que el resto de conductores reales.

Los sistemas de visión por computador han pasado de verse como algo propio a la ciencia ficción a ser una realidad del día a día en los últimos 30 años, gracias en parte a las librerías gratuitas como las OpenCV que han facilitado el acceso a todo el mundo. Como se verá más adelante muchos de los sistemas de ayudas que hay actualmente en los automóviles son gracias a la visión por computador.

En este proyecto se ha decidido usar esos sistemas de visión por ordenador para realizar un programa que sea capaz de detectar y reconocer señales de tráfico, y en un futuro avisar al conductor no sólo de las señales encontradas sino de posibles infracciones que se cometan.

La primera parte del proyecto se basará en la detección de las señales de tráfico que capte la cámara, y la segunda en el reconocimiento del mayor número posible. Todo esto se hará con la ayuda de las funciones que ya vienen implementadas en las OpenCV y sobre el sistema operativo ROS.

2. Problemática actual

En base al Informe mundial sobre la prevención de los traumatismos causados por el tránsito [1] cada día mueren en el mundo más de 3000 personas por lesiones resultantes del tránsito. En los países de ingresos bajos y medianos se concentra aproximadamente un 85% de esas muertes y el 90% de la cifra anual de años de vida ajustado en función de la discapacidad (AVAD) perdidos por causa de esas lesiones. Las proyecciones muestran que, entre 2000 y 2020, las muertes resultantes del tránsito descenderán en torno al 30% en los países de ingresos altos, pero aumentarán considerablemente en los de ingresos bajos y medianos. De no emprenderse las acciones pertinentes, se prevé que, en 2020, las lesiones causadas por el tránsito sean el tercer responsable de la carga mundial de morbilidad y lesiones Figura 1.

1990		2020	
Rango	Enfermedades o traumatismos	Rango	Enfermedades o traumatismos
1	Infecciones de las vías respiratorias inferiores	1	Cardiopatía isquémica
2	Enfermedades diarreicas	2	Depresión unipolar grave
3	Trastornos perinatales	3	Traumatismos causados por el tránsito
4	Depresión unipolar grave	4	Trastornos cerebrovasculares
5	Cardiopatía isquémica	5	Enfermedad pulmonar obstructiva crónica
6	Trastornos cerebrovasculares	6	Infecciones de las vías respiratorias inferiores
7	Tuberculosis	7	Tuberculosis
8	Sarampión	8	Guerras
9	Traumatismos causados por el tránsito	9	Enfermedades diarreicas
10	Anomalías congénitas	10	VIH

Figura 1 – Cambio de rango de las diez causas principales de la carga mundial de morbilidad según los AVAD perdidos [1]

Todas las personas que mueren, se lesionan o quedan discapacitadas por un choque tienen una red de personas allegadas, como familiares y amigos, que resultan profundamente afectadas. En el mundo, millones de personas se enfrentan a la muerte o la discapacidad de familiares debido a lesiones causadas por el tránsito. Sería imposible asignar un valor cuantitativo a cada caso de sacrificio y sufrimiento humano, sumarlos todos y obtener una cifra que refleje el costo social mundial de los choques y las lesiones causadas por el tránsito. Se estima que el costo económico de los choques y las lesiones causadas por el tránsito asciende al 1% del producto nacional bruto (PNB) en los países de ingresos bajos, al 1,5% en los de ingresos medianos y al 2% en los de ingresos altos. El costo mundial se estima en US\$ 518 000 millones anuales, de los cuales US\$ 65 000 millones corresponden a los países de ingresos bajos y medianos; este monto es mayor del que reciben en ayuda al desarrollo.

Enfermedad o traumatismo	Millones de US\$	Rango según los AVAD, 1990	Rango según los AVAD, 2020
VIH/SIDA	919-985	2	10
Paludismo	60	8	-
Enfermedades diarreicas	32	4	9
Traumatismos causados por el tránsito	24-33	9	3
Tuberculosis	19-33	-	7

Figura 2 – Estimación de los fondos destinados mundialmente a investigación y desarrollo sobre determinadas cuestiones [1]

Las lesiones causadas por el tránsito representan una pesada carga no sólo para la economía mundial y de los países, sino también para la de los hogares. La pérdida de quienes ganaban el sustento y el costo añadido de atender a los familiares discapacitados por dichas lesiones sumen a muchas familias en la pobreza. En cambio, se invierte muy poco dinero en prevenir los choques y las lesiones causadas por el tránsito. En la Figura 2 se comparan los fondos gastados en labores de investigación y desarrollo centradas en varios problemas de salud, incluida la seguridad vial. Se gasta relativamente poco en la aplicación de medidas, pese a que muchas intervenciones que permiten prevenir choques y traumatismos se conocen bien, se han probado lo suficiente y son rentables y aceptables para la población.

2.1. Cifras en España

De acuerdo con el informe de siniestralidad vial en España publicado en el 2013 [2] . Las diferentes policías notificaron 89.519 accidentes con víctimas. Estos accidentes ocasionaron 1.680 fallecidos en el momento del accidente o hasta 30 días después del mismo, 10.086 personas fueron ingresadas en un centro hospitalario y 114.634 resultaron heridos leves, según fuentes policiales. Estas cifras, aun siendo elevadas, han supuesto una reducción con respecto al año anterior en el número de fallecidos (-12%) y heridos graves (-3%) y un aumento en los accidentes con víctimas (8%) y en el número de heridos leves (9%). Estos aumentos podrían estar relacionados con la mejora de la notificación especialmente en el ámbito urbano.

El parque de automóviles ha crecido 4.272.351 en el último decenio en todas las categorías de vehículos, y los turismos representan el 68% del mismo. No obstante, en 2013 se observa al igual que ocurrió en el año 2012, un descenso del 1% en la cifra total del parque respecto del año anterior. La antigüedad media del parque de automóviles con menos de 25 años oscila entre los 8,5 años de los tractores industriales y los 11,1 de los camiones y furgonetas, siendo la antigüedad media para los turismos de 9,9 años. Debido a la actual situación económica la antigüedad media de los vehículos está aumentando en los últimos años, esto aumenta el factor de riesgo por dos motivos, los vehículos al ser más antiguos carecen de los sistemas de seguridad más modernos y la antigüedad del propio automóvil es un factor de riesgo en sí, al deteriorarse su mecánica y aumentar la posibilidad de un fallo.

El censo de conductores en el año 2013 con respecto al año anterior ha permanecido estable, siendo la tasa de conductores por 1000 habitantes con edad habilitada para conducir de 666, tasa más elevada que la observada el año anterior. Se sigue observando un envejecimiento paulatino de la población con permiso de conducir, que avanza en paralelo al envejecimiento de la población, el 13,6% de los conductores tienen 65 o más años. Los conductores que cuentan con un permiso inferior a cinco años han pasado del 24% en el 2008 al 17% en el año 2013.

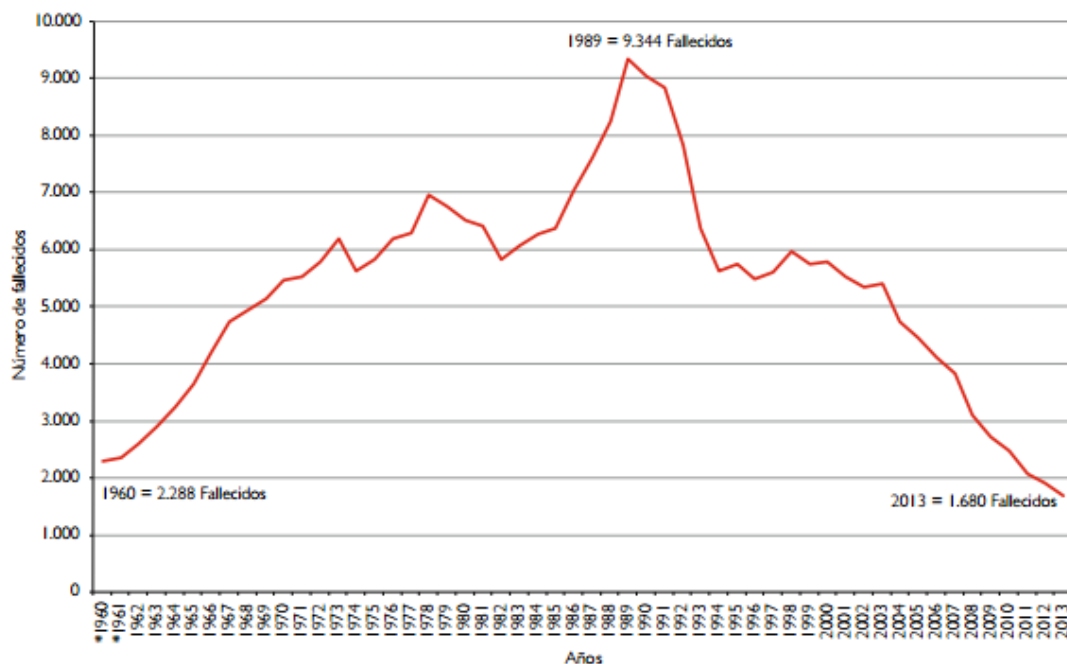


Figura 3 - Evolución de los fallecidos en accidente de tráfico con víctimas. Serie 1960-2013 [2]

Como podemos apreciar, en los últimos 20 años se ha producido un descenso tanto en el número de fallecidos por accidente de tráfico, Figura 3, como en la letalidad de los mismos Figura 4 que ha descendido del 5.2% a un 1.3%. Este descenso es debido a la modernización de la red de tráfico y a las mejoras de seguridad introducidas en los nuevos vehículos.

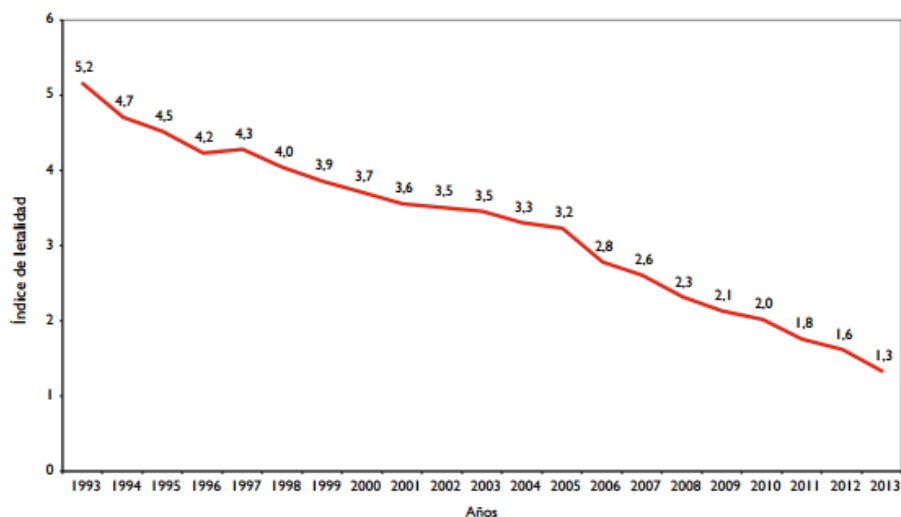


Figura 4 - Evolución de la letalidad (n° de fallecidos/ n° de víctimas $\times 100$) en los accidentes de tráfico con víctimas. Serie 1993–2013 [2]

2.2. Causas

Entre los factores que son a menudo causa de accidentes encontramos:

2.2.1. La velocidad

La velocidad inadecuada, según la valoración del agente en el momento del accidente, estuvo presente en el 10% de los accidentes con víctimas ocurridos en 2013 y este porcentaje se eleva al 17% cuando el accidente con víctimas sucede en vía interurbana.

En el caso de los accidentes mortales se apreció que en un 22% de ellos la velocidad fue un factor concurrente en el accidente. En el año 2013 el total de denuncias de la Dirección General de Tráfico fue de 4.020.262, de estas 2.170.881 estaban vinculadas a la velocidad (54%). Estas denuncias fueron realizadas por la Agrupación de Tráfico de la Guardia Civil y por radares fijos, de tramo y helicópteros.

Vehículos ligeros	Límite 120 km/h		Límite 100 km/h		Límite 90 km/h		Límite 90 km/h	
	Autopistas		Autovías		Vías interurbanas		Vías interurbanas	
	Día	Noche	Día	Noche	Día	Noche	Día	Noche
Velocidad media	118,1	117,4	107,8	107,3	94,2	97,8	92,6	94,6
Percentil 2,5	85	83	65	68	64	66	57	61
Percentil 85	133	130	127	125	115	117	108	110
Percentil 97,5	154	151	145	147	133	137	127	133
% vehículos superan velocidad límite	39,5%	32,6%	27,8%	22,6%	37,2%	25,6%	43,8%	47,5%
% vehículos superan velocidad límite en más 10 km/h	19,4%	16,1%	11,9%	10,0%	21,8%	25,9%	25,5%	28,8%
% vehículos superan velocidad límite en más 20 km/h	8,7%	7,2%	4,4%	4,3%	9,8%	11,9%	12,9%	15,2%
Número de vehículos observados en la muestra	12.356	5.028	38.079	15.456	4.189	1.287	9.205	2.644

Figura 5 -Encuesta para la medición de velocidad libre para los vehículos ligeros. Año 2012 [2]

Como se puede observar un amplio número de vehículos ligeros supera la velocidad máxima permitida en la vía, lo que conlleva un aumento no solo del número de accidentes sino de la letalidad de los mismos.

2.2.2. El alcohol y las drogas

El 12% de españoles que conducen turismos han consumido alguna droga de comercio ilegal y/o alcohol, antes de conducir. Este es el dato general del estudio de prevalencia del consumo de sustancias psicoactivas elaborado por la DGT durante 2013.



Figura 6 - Distribución de los resultados toxicológicos de las pruebas de control de drogas realizadas por la Agrupación de Tráfico de la Guardia Civil, según número de sustancias halladas, en el año 2013. Fuente: DGT [2]

Como se puede observar además en la figura 6 más de la mitad de las personas que conducen bajo los efectos de las drogas lo hacen habiendo consumido más de un tipo distinto, lo que multiplica las posibilidades de sufrir un accidente.

2.2.3. La distracción

La distracción aparece como factor concurrente en un 38% de los accidentes con víctimas. En vía interurbana la distracción se presenta como factor en un 44% de los casos y en las vías urbanas en un 33%. Uno de los motivos que conlleva a la distracción del conductor es la utilización del teléfono móvil durante la conducción. En el año 2013 la Agrupación de Tráfico de la Guardia Civil⁴ formuló 126.345 denuncias por utilización del teléfono móvil. Esta cifra supone un 4% más que en 2012.

2.2.4. Las infracciones

Durante el año 2013 al menos el 49% de los conductores implicados en accidentes ocurridos en vía interurbana había cometido alguna infracción según la opinión del agente informante. Cuando el accidente tiene lugar en vías urbanas al menos el 45% de los conductores cometieron alguna infracción en 2013.

Vía	Infracción	2004	2009	2010	2011	2012	2013
Interurbana	Solo velocidad (Nº)	2.756	3.097	3.194	2.744	2.616	2.657
	% sobre total conductores implicados	4%	5%	5%	5%	4%	4%
	Velocidad y otra (Nº)	11.498	7.096	6.021	5.363	5.103	4.974
	% sobre total conductores implicados	16%	10%	9%	9%	9%	8%
	Conducción distraída (Nº)	13.118	11.206	10.802	10.377	10.282	10.903
	% sobre total conductores implicados	18%	16%	16%	17%	18%	17%
	Circular por sentido contrario (Nº)	198	124	116	139	110	105
	% sobre total conductores implicados	0%	0%	0%	0%	0%	0%
	Adelantar antirreglamentariamente (Nº)	1.407	752	611	502	500	488
	% sobre total conductores implicados	2%	1%	1%	1%	1%	1%
	No mantener distancia de seguridad (Nº)	2.995	2.334	2.285	2.000	1.984	2.135
	% sobre total conductores implicados	4%	3%	3%	3%	3%	3%
	Infracciones de prioridad (Nº)	4.554	2.941	2.776	2.605	2.558	2.749
	% sobre total conductores implicados	6%	4%	4%	4%	4%	4%
	Otras infracciones no de velocidad (Nº)	6.776	7.800	7.746	6.948	6.644	7.773
	% sobre total conductores implicados	9%	11%	12%	12%	11%	12%
	Ninguna / Se ignora (Nº)	30.627	33.001	32.208	29.340	28.620	31.149
	% sobre total conductores implicados	41%	48%	49%	49%	49%	49%

	Total conductores implicados en accidentes con víctimas	73.929	68.351	65.759	60.018	58.417	62.933
	Solo velocidad (Nº)	800	573	443	420	467	426
	% sobre total conductores implicados	1%	1%	1%	1%	1%	0%
	Velocidad y otra (Nº)	5.075	2.196	2.209	1.933	2.010	2.246
	% sobre total conductores implicados	6%	3%	3%	2%	2%	3%
	Conducción distraída (Nº)	13.992	5.604	5.557	5.841	5.826	6.528
	% sobre total conductores implicados	16%	7%	7%	7%	7%	7%
	Circular por sentido contrario (Nº)	383	152	188	182	145	168
	% sobre total conductores implicados	0%	0%	0%	0%	0%	0%
	Adelantar antirreglamentariamente (Nº)	1.278	386	313	309	269	316
Urbana	% sobre total conductores implicados	1%	0%	0%	0%	0%	0%
	No mantener distancia de seguridad (Nº)	2.189	976	1.060	1.092	1.045	1.138
	% sobre total conductores implicados	2%	1%	1%	1%	1%	1%
	Infracciones de prioridad (Nº)	13.296	5.737	5.824	5.898	5.607	6.463
	% sobre total conductores implicados	15%	7%	7%	7%	7%	7%
	Otras infracciones no de velocidad (Nº)	13.370	37.311	36.403	35.895	36.277	22.855
	% sobre total conductores implicados	15%	45%	45%	44%	44%	26%
	Ninguna / Se ignora (Nº)	37.692	30.010	28.806	30.150	30.635	48.934
	% sobre total conductores implicados	43%	36%	36%	37%	37%	55%
	Total conductores implicados en accidentes con víctimas	88.075	82.945	80.803	81.720	82.281	89.074

Figura 7 - Infracciones de los conductores implicados en accidentes de tráfico en vías interurbanas y urbanas. Años 2004, 2009, 2010, 2011, 2012, 2013 [2]

El tipo de infracciones que más podría ayudar a reducir un sistema de detección y reconocimiento de señales de tráfico son los siguientes:

- Solo velocidad
- Velocidad y otra
- Adelantar antirreglamentariamente
- Infracciones de prioridad

Es decir, un 28% de las infracciones en vía interurbana y un 22% de las producidas en vías urbanas podrían verse reducidas mediante un sistema de este tipo.

3. Estado del arte

En la actualidad se están desarrollando distintos sistemas de seguridad muy novedosos, basados en la visión por ordenador, en laser, o en otro tipo de sensores electrónicos. En base a [3] y [4] se comentan a continuación algunos ejemplos de los mismos:

3.1. Detección de sueño o falta de atención al volante

En algunos turismos de gama media-alta (no necesariamente de lujo) empieza a verse este tipo de sistema incluso de serie (por ejemplo en el Volkswagen Passat). Puede tener varios nombres (por ejemplo detección de fatiga) pero en el fondo las diferentes versiones consisten en lo mismo, intentar detectar si el conductor no está en óptimas condiciones para seguir conduciendo.

Normalmente son un sistema electrónico con un sensor en el volante, que cuenta cuántas veces por minuto el conductor realiza pequeñas correcciones en la dirección. Se sabe que para mantenernos en el carril, los conductores no mantenemos el volante quieto y fijo, sino que corregimos casi constantemente dos o tres grados hacia la derecha o hacia la izquierda, para intentar ir lo más centrados posible en él.

Recientemente se están desarrollando sistemas que detectan la fatiga y el sueño midiendo el número de parpadeos y la duración de los mismos. De esta forma se puede saber si el conductor del vehículo está cansado o cerca de quedarse dormido.

3.2. Detección de cambio de carril involuntario

Algunos fabricantes utilizan dos sensores seguidores de línea, uno a cada lado del coche, en el paragolpes delantero, muy cerca del suelo, que reconocen y “siguen” las líneas blancas, continuas o discontinuas, de la carretera.

Otros fabricantes optan por una cámara colocada en lo alto del parabrisas (por el interior, más o menos delante del espejo retrovisor) y reconocen las líneas que marcan el carril por el que circulamos. Tanto en uno como en otro, un microprocesador está permanentemente atento a que la trayectoria del coche se mantenga entre las dos líneas (izquierda-derecha).

Si en algún momento pisamos o sobrepasamos alguna de las líneas, y no habíamos activado el intermitente (luz indicadora de cambio dirección) del lado correspondiente, el sistema interpreta que el cambio es involuntario (por un despiste) y actúa. Algunos sistemas solo advierten al conductor, normalmente con una vibración en el asiento o bien en el volante, pero también puede sonar una pequeña alarma sonora, y también un mensaje en la pantalla digital del cuadro de instrumentos.

Otros sistemas un poco más avanzados, actúan sobre la dirección, y hacen girar el volante ligeramente en la dirección opuesta, para corregir la deriva en la trayectoria que hace que pisemos la línea en cuestión. Cada vez más sistemas de dirección asistida utilizan un servomotor eléctrico, así que no es complicado utilizar este motor para hacer girar el volante.

Estos sistemas, por el momento, no mantienen el coche en el carril, solo corrigen un poco el volante, pero si la deriva es excesiva, o el conductor insiste aplicando más fuerza sobre el volante, el sistema se vence.

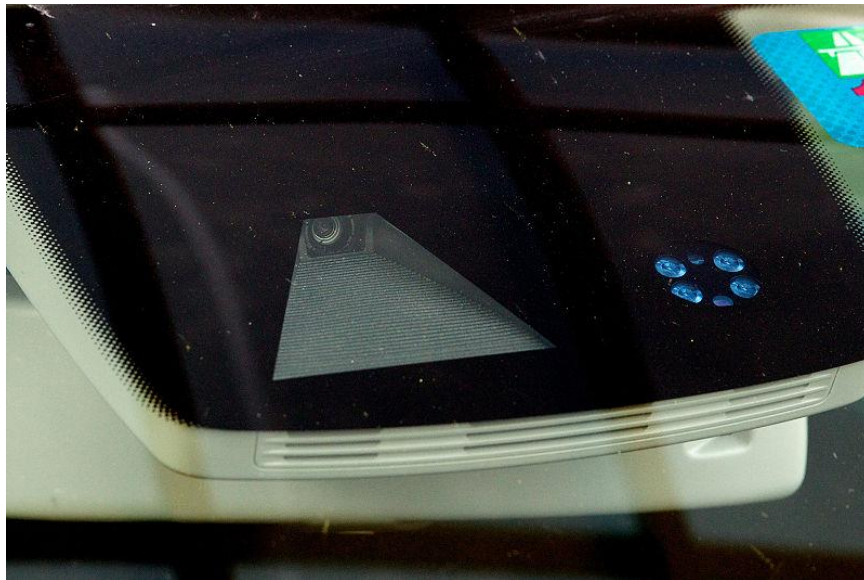


Ilustración 1 – Cámara del sistema de detección de cambio de carril de un VW Golf [4]

3.3. Advertencia de ángulo muerto en el espejo retrovisor

Algunos coches montan (de nuevo como opción) un sistema que avisa de vehículos en el ángulo muerto, mediante una señal visual sobre el espejo retrovisor, o en el marco del espejo (suele ser un led amarillo que parpadea, o un pequeño triángulo de peligro). El sistema emplea radares (es decir un sistema de emisión de ondas de radio), en las esquinas del paragolpes trasero (o en el lateral del coche, a veces en el mismo espejo retrovisor) orientados hacia el área que queda oculta. Por ejemplo es opcional en el Audi A6.

3.4. Aparcamiento automático

El aparcamiento automático es un sistema de maniobra autónoma que mueve un vehículo de un carril de tráfico a un hueco de aparcamiento para realizar un estacionamiento en paralelo o batería. El sistema de aparcamiento automático tiene como objetivo mejorar la comodidad y la seguridad de la conducción en entornos restringidos donde se requiere mucha atención y experiencia para dirigir el coche. La maniobra de aparcamiento se consigue por medio de un control coordinado del ángulo de dirección y la velocidad que tiene en cuenta la situación real en el medio ambiente para asegurar el movimiento libre de colisiones dentro del espacio disponible.

El coche es un ejemplo de un sistema no holonómico donde el número de comandos de control disponibles es menor que el número de coordenadas que representan su posición y orientación. Se ha demostrado que el sistema de aparcamiento automático mejora la comodidad y la seguridad al reducir el nivel de estrés que la gente siente cuando la realiza la maniobra manual de aparcamiento.

3.5. Reconocimiento de objetos

Este sistema pretende evitar sobre todo atropellos de peatones o ciclistas. El sistema ContiGuard quiere ir un paso más allá y por el momento es algo experimental. Dos cámaras de alta resolución, colocadas en la parte alta del parabrisas, permiten tener una visión estereográfica al microprocesador del sistema.

El sistema es capaz de distinguir entre peatones, ciclistas, coches y otros objetos, medir la distancia hasta ellos, y predecir su trayectoria, verificando si esta se corta con la que lleva el coche, y determinar por tanto si existe riesgo de un accidente, para tomar las medidas preventivas conducentes a evitarlo o aminorar las consecuencias. Advertirá al conductor, e incluso podría accionar automáticamente los frenos del vehículo.

El otro sistema que se está probando (en Alemania, bajo el nombre Amulett Car2X) funciona por ondas de radio. Los peatones y ciclistas deberían llevar un pequeño transpondedor, muy similar al sistema RFID, que incluso no necesitaría alimentación eléctrica (si es de tipo pasivo).

El coche debe llevar un emisor-receptor y recibe la señal de respuesta de los transpondedores. Lo bueno de este sistema es que los peatones podrían estar ocultos (por ejemplo niños detrás de un vehículo alto estacionado) pero el microprocesador del coche sabe que están ahí y advierte de ello al conductor.



Ilustración 2 –Funcionamiento del sistema de reconocimiento de objetos [4]

3.6. Control de descenso de pendientes

El control de descenso de pendientes permite un descenso de pendientes suave y controlado en terrenos irregulares sin que el conductor necesite tocar el pedal del freno. Cuando está activado, el vehículo descenderá mediante el sistema de frenos ABS para controlar la velocidad de cada rueda. Si el vehículo acelera sin intervención del conductor, el sistema aplicará automáticamente los frenos para reducir la velocidad a la deseada. Los botones del control de crucero permiten ajustar la velocidad a un nivel cómodo. La aplicación de presión al acelerador o el freno de pedal anularán el sistema cuando el conductor lo requiere.

Con el control de descenso de pendientes los conductores pueden estar seguros de que incluso el trayecto cuesta abajo con terreno resbaladizo o rugoso será suave y controlado, y que será capaz de mantener el control mientras exista suficiente tracción.

3.7. Detección y reconocimiento de señales de tráfico

Los sistemas comentados son sólo algunos ejemplos de las tecnologías que se aplican hoy en día a la seguridad del automóvil. Sin embargo este proyecto se centra en la detección y el reconocimiento de señales de tráfico.

Los sistemas de reconocimiento de señales de tráfico se incluyen dentro de las características comúnmente llamadas ADAS (Advanced driver assistance systems). Esta tecnología ha sido desarrollada por empresas como Ayonix y Continental. Apareció por primera vez a finales de 2008 sobre el nuevo diseño del BMW 7-series, y el siguiente año en el Mercedes-Benz S-Class. Actualmente, estos sistemas sólo detectan los límites de velocidad. Aunque los sistemas de la segunda generación ya detectan otras restricciones, como es el caso del Volkswagen Phaeton 2011.

Mediante un tratamiento digital de imágenes que capta una cámara que va grabando la carretera, un ordenador es capaz de leer una señal, interpretarla y ayudar al conductor para aumentar su seguridad o incluso ahorrarse multas por distracción. La tecnología TSR (Traffic Sign Recognition) tiene una aplicación pasiva y activa.

- **Aplicación pasiva:** Esta aplicación consiste en advertir al conductor mediante pictogramas, sonidos o cualquier otro tipo de señales de que está infringiendo una limitación de velocidad, que entra en zona de curvas peligrosas, peligro de nieve / hielo, etc. El conductor tiene como obligación leer todas las señales, pero puede ser de ayuda ante una distracción.
- **Aplicación activa:** Esta aplicación consiste en que el coche interviene para evitar un accidente. Por ejemplo, si se circula a 90 Km /h y el coche lee un STOP, y no interpreta que el conductor vaya a parar, ordena a los frenos a parar el coche antes de llegar a la señal para evitar un posible accidente. Dependiendo del fabricante, esta ayuda puede ser desconectable voluntariamente o no.

Actualmente esta tecnología ya está disponible en algunos coches:

- Audi A8
- BMW Serie 7
- BMW Serie 5 Gran Turismo
- BMW Serie 5
- Mercedes-Benz Clase E
- Mercedes-Benz Clase S
- Opel Insignia
- Opel Corsa
- Saab 9-5
- Volkswagen Phaeton
- Lexus GS
- Lexus LS

Marcas que ya utilizan estos sistemas serían: Audi, BMW, Citroen, Ford, Honda, Infiniti, Jaguar, JEEP, Land Rover, Lexus, Mercedes, Nissan, Opel, Peugeot, Porsche, Renault, Toyota, Volkswagen o Volvo-

En la actualidad existen también numerosas publicaciones científicas acerca de la detección y el reconocimiento de señales de tráfico y que tratan ampliamente estos temas. Desde las que estudian por ejemplo el espacio de colores que sería mejor para realizar la detección de imágenes [5] hasta las que entran en discusión sobre que métodos serían los mejores para reconocer señales [6], redes neuronales, SURF, etc.

4. Descripción general del sistema

El proyecto cuenta tanto con elementos hardware, la cámara que transmitirá las imágenes al ordenador, el coche donde va montada y el ordenador donde se ejecuta. Como con elementos de software, el lenguaje en el que se programa (C++), el entorno de trabajo (QtCreator), el conjunto de librerías usadas (OpenCV) y el framework sobre el que se realizará la programación (ROS).

4.1. Introducción a la aplicación

Al comenzar el desarrollo del proyecto se hizo un estudio inicial sobre los objetivos a alcanzar, la forma de desarrollarlo y los problemas que podrían surgir durante la realización del mismo.

Hay diferentes tipos de señales, precaución, obligación, prohibición..... Todas ellas se pueden distinguir fácilmente a simple vista ya que tienen colores y/o formas distintas. Así nos encontramos por ejemplo con señales triangulares con borde rojo indicando precaución, señales circulares con fondo azul señalando obligación o señales circulares con borde rojo para una prohibición. En principio el proyecto quería abarcar todas las posibles, pero finalmente se ha decidido centrar en las señales de prohibición y precaución y en futuras ampliaciones añadir el resto.

También nos encontramos con dos entornos distintos para las señales, las vías interurbanas y las urbanas, cada con sus pros y sus contras para el funcionamiento del programa. En las vías interurbanas la velocidad es notablemente superior y por tanto se disponen de menos frames que analizar, esto es perjudicial porque cuantas más imágenes se adquieran de una señal más aumentamos la probabilidad de éxito, no obstante al contrario que en las vías urbanas el entorno es bastante constante, y no es frecuente que haya elementos que oculten las señales, ni hay tantas sombras como las producidas en la ciudad por los edificios y que dificultan en gran medida la detección. Se decide por tanto tratar de hacer funcionar el programa en ambos tipos de entorno.

También hay que tener en cuenta que pese a que las señales se suponen internacionales, hay ligeras diferencias entre países, por ejemplo en los números que marcan las velocidades. En la detección no afecta ya que se segmenta la imagen por color y formas, pero a la hora de reconocerlas si puede ser un problema, por lo que se realiza el programa sólo con señales del territorio español.

Así pues se decide centrar el programa en detectar todas las señales triangulares y circulares con borde rojo, y realizar el reconocimiento de:

- Prohibido superar 100 km/h
- Prohibido superar 90 km/h
- Prohibido superar 70 km/h
- Prohibido superar 60 km/h
- Prohibido superar 40 km/h
- Prohibido adelantar
- Prohibida la circulación a camiones
- Ceda el paso
- Precaución glorieta

4.2. Hardware

4.2.1. Cámara SJ4000

La cámara SJ4000 [7] es una cámara deportiva de acción similar a la GoPro. Una de sus principales características es su lente, que capta en un ángulo de 170°, esto permite captar todo el ancho de la carretera, y sería muy útil también en otras aplicaciones como detectar adelantamientos. La cámara tiene una lente gran angular de ojo de pez que capta por tanto las imágenes distorsionadas, esto hace que sea necesaria una calibración de la cámara para poder procesar de forma correcta las imágenes.



Ilustración 3 - Cámara SJ4000 [7]

La cámara también dispone de una conexión WiFi para conectarla al ordenador sin necesidad de usar cables, finalmente se decidió usar la conexión por cable con el ordenador por razones que detallaremos más adelante.

El resto de características principales de la cámara se citan a continuación:

- Sensor de imagen Aptina AR0330
- Conexión WiFi
- Panel LCD 1.5
- Resolución de imagen: de 12 megapíxeles (4032 * 3024) / 10 mega píxeles (3648 * 2736) 8 mega píxeles ((3264 * 2448) / 5 megapíxeles (2592 * 1944) 3 MP (2048 * 1536) / 2MHD (1920 * 1080) / VGA (640 * 480) / 1,3 M (1280 * 960)
- Resolución de vídeo: 1920 * 1080 30fps / 1280 * 720 60fps / 848 * 480 60fps / 640 * 480 60fps
- Formato de compresión de vídeo H.264
- Imagen / formato de archivo de vídeo Formato de vídeo: MOV formato / imagen
- Zoom digital x4
- Lente gran angular de lente de ojo de pez de 170 ° de alta definición, no deformable
- Conexiones USB 2.0, HDMI
- Duración de la batería 80 minutos (1080P)
- Batería extraíble de 900mAh
- Dimensiones 29.8x59.2x41 mm
- 44g Peso (sin baterías), 58g con las baterías

Conexión de la cámara

La cámara tiene la posibilidad de conectarse al ordenador mediante WiFi, lo que facilitaría su instalación en el coche, evitando tener que estar usando un cable.

La conexión mediante WiFi se logró hacer funcionar en el reproductor del VLC, no obstante a la hora de reproducir mediante el programa no daba una imagen continua, sino que aparecía y desaparecía constantemente. A pesar de los numerosos intentos de hacer la conexión evitando el uso de cables al final no fue posible. La cámara SJ4000 transmite la imagen mediante *rtsp* y en formato MOV, se acabó concluyendo que los codecs de los que se disponía no podían reproducirlo.

Finalmente y ante la imposibilidad de usar el WiFi se optó por la conexión mediante el cable USB.

4.2.2. IVVI 2.0

IVVI 2.0 es una plataforma de investigación Laboratorio de Sistemas Inteligentes [8] para la implantación de sistemas inteligentes basados en visión por computador y las técnicas de láser, con el objetivo de desarrollar y probar tecnologías ADAS. El propósito del IVVI es probar nuevos algoritmos en condiciones reales. En este momento, están siendo investigados diferentes métodos de detección para carriles de tráfico, peatones, vehículos, señales de tráfico, vigilancia del conductor y de posicionamiento. Pueden ser tomados como entradas para algunas ADAS como el Sistema de permanencia en carril, Control de cruce adaptativo, Protector de peatones, Reconocimiento de señales de tráfico, Sistemas de piloto automático. Todos los ADAS diseñados trabajan en entornos urbanos y de carretera, y proporcionan información en tiempo real para el conductor.

Los diferentes dispositivos de detección incluyen una cámara CCD a color Figura 8 montado en el parabrisas, para la detección de tráfico signos y otras señales verticales; un sistema de visión estéreo para detección del carril carretera y clasificación y reconocimiento de objetos (vehículos, peatones, etc.) en condiciones de conducción diurnas se ha incluido; una cámara de infrarrojos montada en el espejo retrovisor para la detección de peatones en condiciones de conducción nocturna; un sistema Kinect, colocado en el tablero de instrumentos, para la detección de rostros y monitoreo del conductor. Un láser de múltiples capas está montado en el parachoques delantero para la detección de objetos, y, finalmente, los dispositivos generales se completan con un receptor GNSS y una IMU, que se integran en una plataforma en el techo del vehículo IVVI y alineado con el eje trasero.



Figura 8 – Dispositivos a bordo del IVVI [8]

El sistema de procesamiento se basa en un ordenador colocado en el maletero del vehículo, que se utiliza para el procesamiento de toda la información sensorial en condiciones de tiempo real. Con el fin de proporcionar información al conductor está conectado a los altavoces del vehículo y diferentes mensajes de advertencia de audio y alertas son emitidos. Además, una pantalla en el salpicadero del conductor muestra advertencias visuales y alertas de peligro.

En el vehículo se encuentran disponibles los datos del dispositivo directamente desde sensores a las aplicaciones o datos de bajo nivel también pueden adquirirse a través de ROS. La ventaja de la sincronización de datos de bajo nivel a través de ROS es la marca de tiempo de adquisición de datos. Por lo tanto, se ha mejorado la plataforma IVVI de adquisición de datos independiente para la adquisición de datos sincronizada utilizando la arquitectura ROS. Es decir, ROS permite una colección de drivers y middleware que pretende simplificar la compleja tarea de la adquisición de datos y la sincronización del sensor. Así, las aplicaciones presentadas en este trabajo se han creado en consecuencia para ROS (

Figura 9 (a)). Además, los procesos de fusión sensor permiten para mejorar el rendimiento de cada aplicación en alto nivel etapas.

Así, en la arquitectura diseñada, todos los procesos se comunican entre sí con el fin de perfeccionar la información y el conocimiento, proporcionando mayor información para mejorar la toma de decisiones del proceso (es decir: para advertir de forma segura el conductor de una posible colisión). La arquitectura ROS permite la comunicación entre procesos en una organización independiente y forma modular, permitiendo ordenador de a bordo ejecutar múltiples algoritmos paralelos con el fin de lograr ambas tareas de bajo nivel (Por ejemplo: adquisición de sensor, y los datos de preprocesamiento, como se muestra en la Figura 9 (b)), y el alto nivel objetivos como la detección de peatones. Por otra parte, la escalabilidad de ROS permite adaptar todo el sistema a cambios de tecnología dentro del vehículo debido a nuevos sensores o superior capacidades de las aplicaciones, es decir, el uso de mensajes estándar de los sensores del mismo tipo hace que todo el sistema sea inmune a los cambios de hardware.

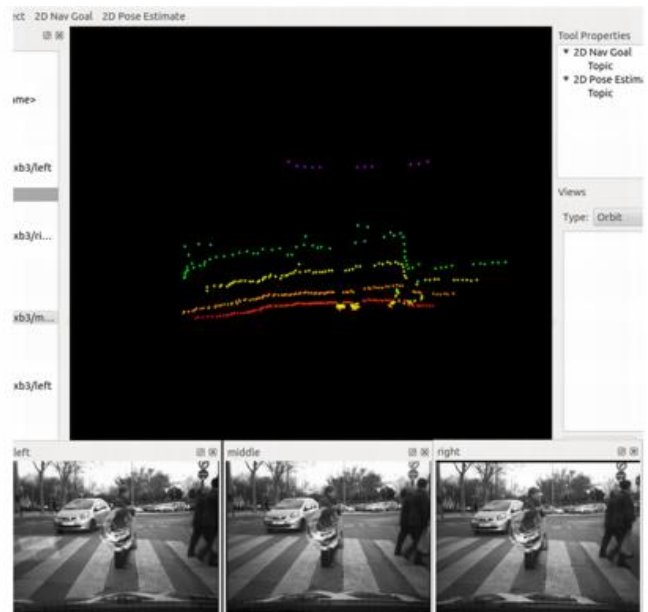
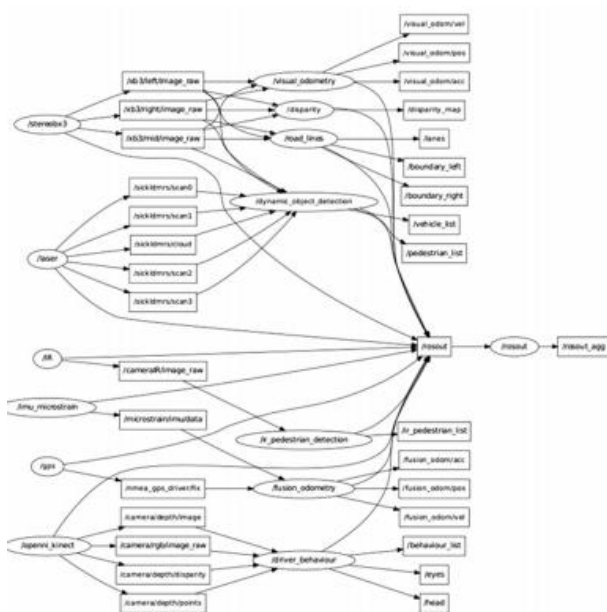


Figura 9 – (a) imagen de la izquierda, diagrama de la adquisición global de datos y la sincronización de procesos a nivel bajo; (b) imagen derecha, ejemplos de la adquisición en tiempo real de imágenes mediante el escáner láser y la cámara estéreo. [8]

4.3. Software

4.3.1. QtCreator

El entorno de desarrollo que se ha decidido usar para la programación es QtCreator. Qt Creator es un entorno de desarrollo integrado C ++ multiplataforma, JavaScript y QML. Qt Creator utiliza el compilador de C ++ de la colección de compiladores de GNU con Linux.

Fue elegido porque es el IDE con el que estoy más familiarizado a trabajar en la universidad, además es software abierto y hay disponible gran cantidad de información sobre cómo trabajar con ROS y con OpenCV usando QtCreator, lo que facilitó el proceso de instalación inicial del software.

4.3.2. OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Las funciones de las librerías de OpenCV se centran principalmente en aplicaciones para visión por ordenador, por lo que son ideales para este proyecto. Se decidió usar OpenCV porque muchas de sus funciones que vienen ya programadas nos pueden servir en nuestra aplicación y ahorrar así una gran cantidad de tiempo que se perdería programando todo desde cero.

4.3.3. ROS

Es un framework, basado en Ubuntu, para el desarrollo de software para robots. ROS provee los servicios estándar de un sistema operativo tales como abstracción del hardware, control de dispositivos de bajo nivel, implementación de funcionalidad de uso común, paso de mensajes entre procesos y mantenimiento de paquetes. Está basado en una arquitectura de grafos donde el procesamiento toma lugar en los nodos que pueden recibir, mandar y multiplexar mensajes de sensores, control, estados, planificaciones y actuadores, entre otros.

Como se dijo en la descripción del IVVI 2.0 el uso de ROS facilita una gran cantidad de tareas y permite abstraerse del hardware. Permite trabajar mediante nodos que se comunican entre ellos transmitiendo mensajes en topics a los que se pueden suscribir otros nodos. ROS cuenta además con paquetes que pueden facilitar el desarrollo de la aplicación como un nodo que procesa la imagen a través de la matriz de calibración que se le proporcione y la deja lista para ser usada.

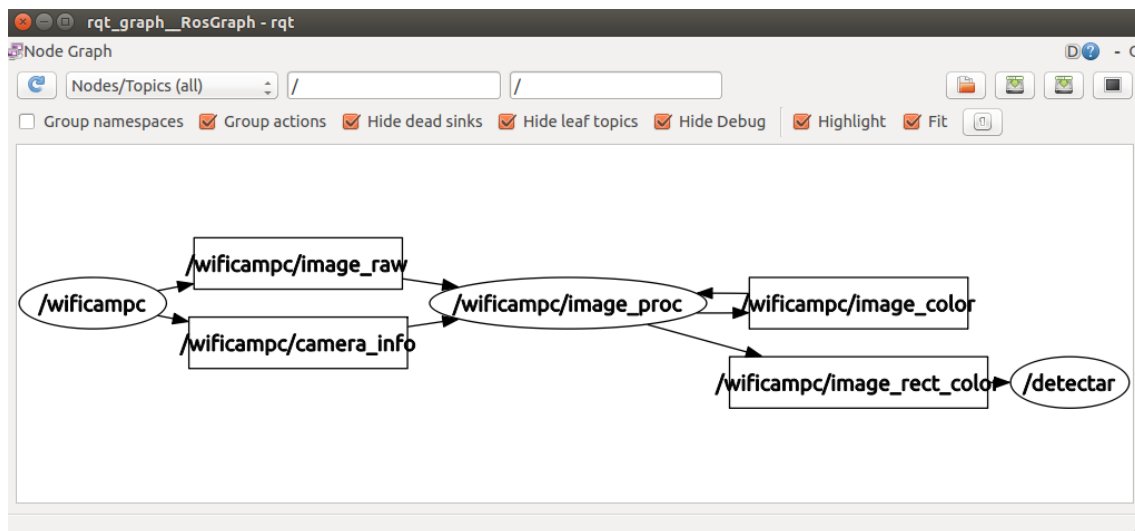


Figura 10 – Lista de topics y nodos ejecutándose durante el programa

En la Figura 10 podemos observar las relaciones entre los nodos y los topics usados para comunicarlos. Como se ha descrito antes tenemos un nodo encargado de captar las imágenes de la cámara (*wificampc*) y que usa dos topics *image_raw* y *camera_info* en los que publica las imágenes que capta (sin rectificar aún) y la información relativa a los parámetros de la cámara. A la información publicada por *wificampc* se suscribe *image_proc*, que se encarga de rectificar la imagen de acuerdo a sus parámetros y publicar esta imagen en un nuevo topic llamado *image_rect_color*. Finalmente a las imágenes publicadas en *image_rect_color* se suscribe en nodo *detectar*, encargado de procesarlas para detectar y reconocer las señales que haya en esa imagen.

5. Estructura del programa

El programa se podría dividir de distintas formas. Una de ellas podría ser en nodos, y constaría de varios. Uno primero que capta las imágenes de la cámara y las publica, otro nodo a continuación que la transforma a partir de la matriz de calibración, corrige la distorsión provocada por el ojo de pez y a continuación la pública de nuevo en un topic. Y finalmente habría un último nodo que se encargaría de la detección y el reconocimiento de la imagen.

No obstante esta estructura no da una idea precisa del funcionamiento real del programa. Por funciones podríamos dividirlo de otra manera más amplia, aunque parte coincida con la anterior.

Primero, como antes, estaría la transmisión de la imagen captada por la cámara. A continuación se transforma la imagen a partir de la matriz de calibración previamente calculada y corrección de la distorsión provocada por la lente de la cámara tipo pez para que la imagen quede como si fuera captada por una lente normal.

A continuación se lleva a cabo la detección de las señales en la imagen. Para ello inicialmente se segmenta la imagen por colores, filtrando el color rojo del resto (como veremos más adelante se han tenido que bajar bastante los umbrales lo que ocasiona falsos positivos en ocasiones). Una vez que se ha segmentado la imagen por colores se realiza un pos-procesamiento para eliminar la gran cantidad de ruido que ha quedado, y de partes que han pasado el filtro de color que por su forma no pueden ser señales pero pueden confundirse con una.

Una vez la imagen ha quedado filtrada se procede a buscar triángulos y círculos en la imagen post-procesada. La zona de la imagen donde se detecte un círculo o un triángulo se guarda como una región de interés (ROI) donde a continuación se pasará reconocer de qué señal se trata.

Finalmente cada ROI que se detecta es pasada por los clasificadores previamente entrenados, que son los que se encargan de decidir la señal a la que corresponde, y mandan un mensaje a la terminal informando de ello.

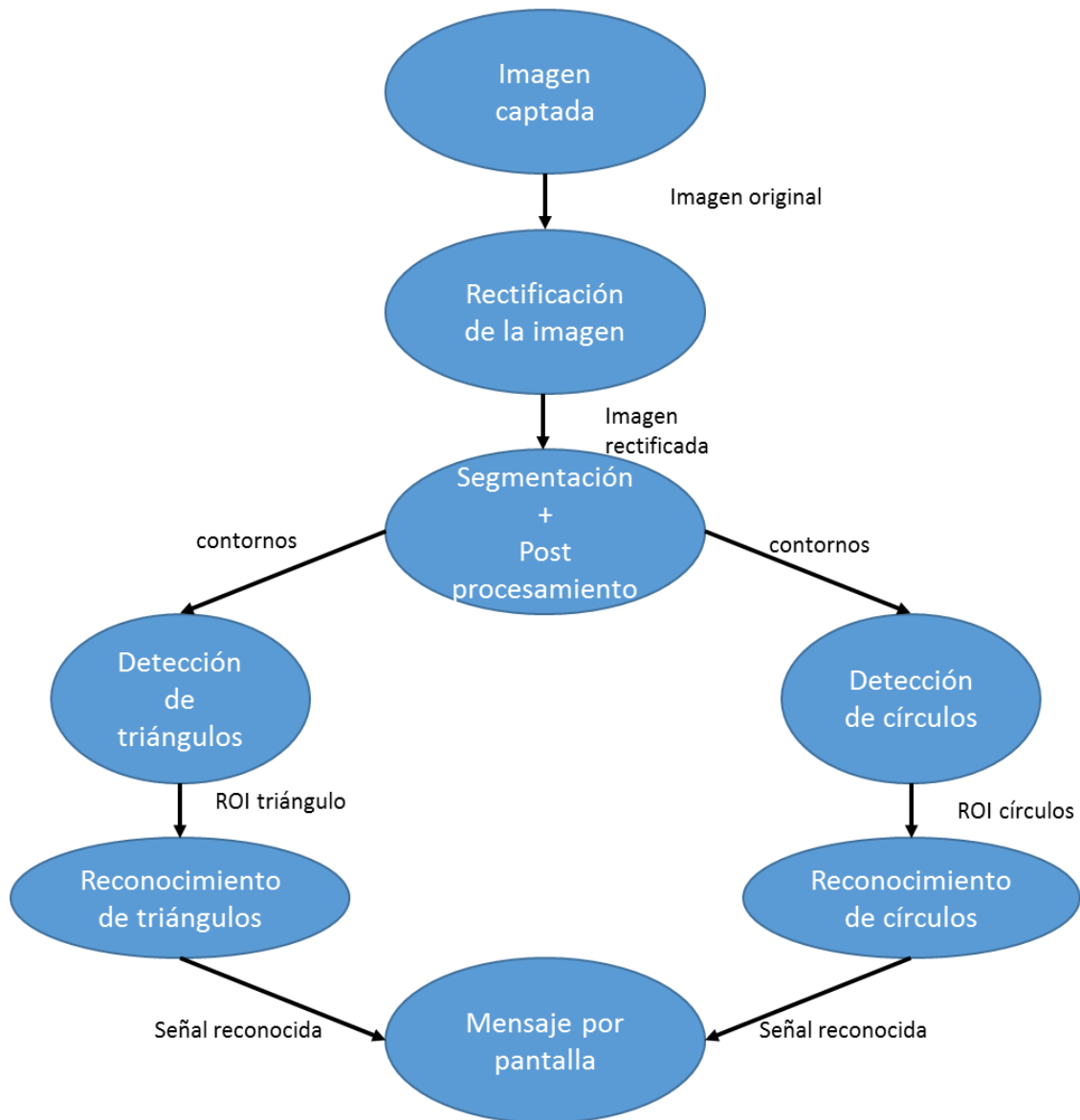


Figura 11 – Estructura del programa

6. Calibración de la cámara

Como se dijo al describir las características de la cámara, su lente tipo pez provoca una distorsión en la imagen que provoca que líneas rectas se vean curvadas Figura 12. Esto puede afectar a la detección de señales puesto que los círculos y triángulos ya no serían exactamente como esperan las funciones encargadas de localizarlos y aumentaría el número de fallos.

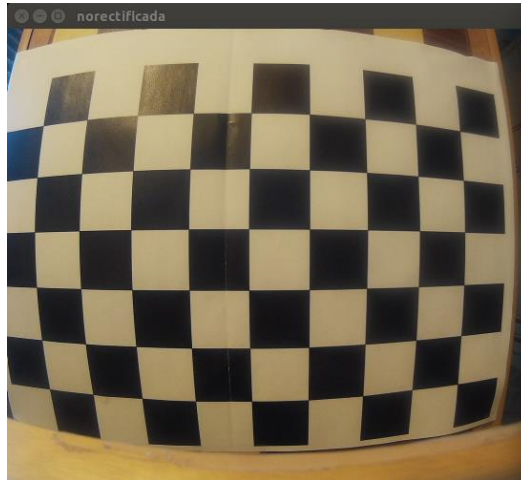


Figura 12 – Tablero de calibración sin rectificar

El proceso de calibración se hará siguiendo el modelo de una cámara tipo pinhole [9]. Según este modelo una imagen se forma proyectando los puntos en 3D a un plano de imagen usando una transformación de perspectiva.

$$s \mathbf{m}' = \mathbf{A}[\mathbf{R}|\mathbf{t}]\mathbf{M}'$$

o:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

- (X, Y, Z) son las coordenadas de un punto de un punto 3D en el espacio exterior de coordenadas
- (u, v) son las coordenadas del punto de proyección en unidades de pixel.
- \mathbf{A} es la matriz de la cámara, o matriz de parámetros intrínsecos.
- (c_x, c_y) es un punto principalmente, normalmente el centro de la imagen.
- f_x, f_y son las distancias focales expresadas en unidades de pixel.

Estos parámetros una vez calculados se mantienen fijos independientemente de la escena captada, sólo cambiarían si por ejemplo cambian las distancias focales al variar el zoom de las lentes, que en nuestro caso se mantiene constante.

La matriz $[R|t]$ es la matriz de parámetros extrínsecos, se usa para describir el movimiento de la cámara, ya sea de rotación o de traslación alrededor de una escena estática, o viceversa.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$u = f_x * x' + c_x$$

$$v = f_y * y' + c_y$$

No obstante la mayoría de las lentes en la realidad están distorsionadas, ya sea radial o tangencialmente, por lo que el modelo exterior pasaría a ser considerado de la siguiente forma:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1 r^2+k_2 r^4+k_3 r^6}{1+k_4 r^2+k_5 r^4+k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$

$$\text{where } r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

Donde k_1, k_2, k_3, k_4, k_5 , y k_6 son coeficientes de distorsión radial, y p_1 y p_2 lo son de distorsión tangencial.

Nuestra matriz de distorsión calculada solo cuenta con los parámetros k_1, k_2, p_1 y p_2 . Los coeficientes de distorsión se mantienen constantes respecto a la escena captada, pero además y al contrario que los parámetros intrínsecos que necesitan ser modificados ante cambios de escala, éstos se mantienen constantes a pesar de modificar la resolución de la imagen.

En el caso de la SJ4000 se trata de la rectificación de una cámara monocular con corrección de distorsión solamente. Para transformar un píxel de la imagen de entrada en uno en la imagen de salida, se envía a través de la serie de transformaciones $K - D - R - K'$ Figura 13 – Proceso de rectificación de la imagen captada por la cámara. K . En este caso, puesto que no hay rotación, traslación, o la ampliación de la imagen original, $K = K'$, y sólo son necesarios los elementos de K y D .

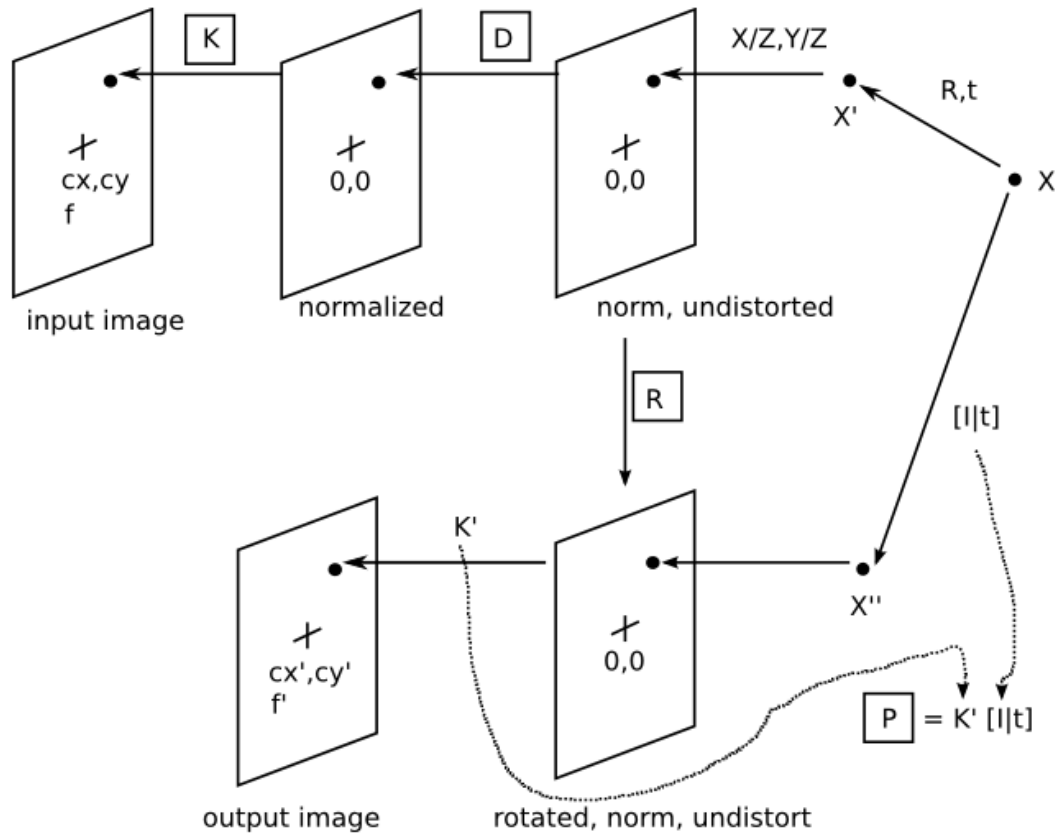


Figura 13 – Proceso de rectificación de la imagen captada por la cámara [10]

El cálculo de estas matrices que permiten la calibración de la imagen se lleva a cabo mediante un paquete de ROS llamado *camera_calibration* [10]. Para ello se utiliza un “tablero de ajedrez” Figura 12, que se mueve frente a la cámara, variando las distancias y los ángulos del mismo. Según va adquiriendo imágenes se va completando la calibración de la cámara, a continuación se comprueba la imagen y si se considera que la calibración es correcta se guardan los parámetros y se convierten a formato YML, para que sean entendibles por ROS. Estos datos de calibración de la cámara se tienen que cargar en el parámetro *camera_info_url*.

No obstante con cargarlos en ese parámetro no es suficiente, eso no produce una imagen sin distorsión. Para obtener la imagen rectificada se necesitan publicar estos parámetros y la imagen original en 2 topics, *camera_info* e *image_raw*, a los que se suscribirá un nuevo paquete de ROS llamado *image_proc*.

Image_proc obtendrá la imagen captada directamente de la cámara suscribiéndose a *image_raw* y le aplicará los parámetros de la misma que se publiquen en *camera_info*. Aplicada esta transformación *image_proc* publicará varios topics con la imagen rectificada a color, en blanco en negro, sin rectificar.....El que interesa para realizar luego la detección sería *image_rect_color*, por lo que es al que se suscribirá el siguiente nodo de nuestro programa.

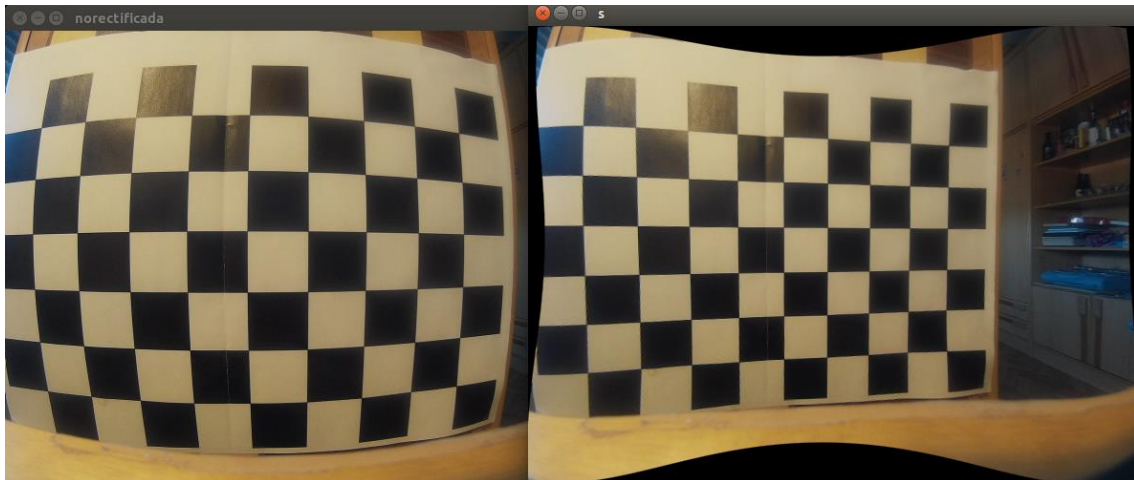


Figura 14 – Comparación de la imagen original (12-a, izquierda) con la rectificada (12-b, derecha)

En la Figura 14 podemos ver los resultados de la calibración de la cámara. En la imagen sin calibrar se puede observar como la lente distorsiona las líneas que deberían ser rectas y las deforma hacia afuera. Tras el proceso de calibración la imagen queda como la percibiría un ojo humano, se puede apreciar que las líneas de la cuadrícula que antes aparecían curvadas ahora son completamente rectas.

```
# oST version 5.0 parameters

[image]

width
1280

height
720

[narrow_stereo]

camera matrix
1834.869175 0.000000 640.273710
0.000000 1369.458501 363.201069
0.000000 0.000000 1.000000

distortion
-1.679203 2.655395 -0.034977 -0.010689 0.000000

rectification
1.000000 0.000000 0.000000
0.000000 1.000000 0.000000
0.000000 0.000000 1.000000

projection
1309.391846 0.000000 624.203727 0.000000
0.000000 992.949280 335.963736 0.000000
0.000000 0.000000 1.000000 0.000000
```

Figura 15 – Matriz de calibración resultante

7. Detección de señales

7.1. Espacio de colores

Se tuvieron en cuenta varios espacios de colores a la hora de decidir con cual trabajar para segmentar la imagen. De acuerdo con [5] el mejor a la hora de trabajar con señales de tráfico es CBM (Color Barycenter Model) sin embargo complicaba en gran parte el funcionamiento del programa, ya que no hay funciones de Opencv con CBM y la mayoría de investigaciones se han realizado trabajando con otros espacios, por lo que finalmente se descartó su uso y eligió usar HSV.

Se decidió utilizar HSV sobre por ejemplo RGB, CMYK o YCrCb puesto que es más inmune a los cambios de iluminación y están más claros los rangos del color rojo bajo distintas condiciones lumínicas.

El Espacio de colores HSV se divide en 3 componentes (Hue/Matiz, Saturation/Saturación, Hue/Valor).

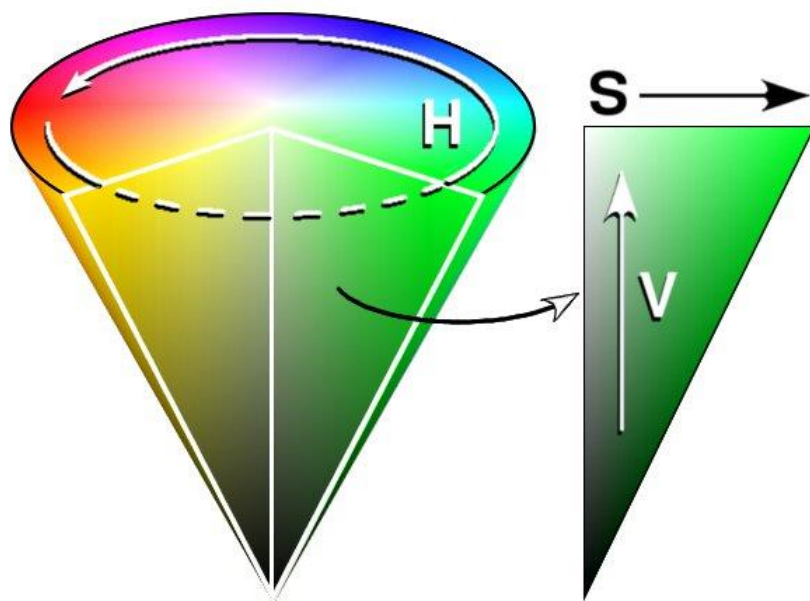


Figura 16 - Espacio de colores HSV

En la *Figura 16* podemos ver el significado de cada componente. El matiz representa una circunferencia que puede tener valores desde 0 a 360° (en OpenCV al ser representado por un valor de 8 bits solo alcanza 0-255, por lo que la circunferencia queda representada en el rango de 0 a 179).

El color rojo esta justo en el 0, por lo que se puede extender por los valores más altos y los más bajos del matiz, sin embargo experimentalmente se ha comprobado que los valores que toman las señales están en el rango de 115-180

La saturación representa la pureza de un color, cuanto más bajo sea el valor más blanco será el color, mientras que al aumentar el color se aproxima más al definido por el matiz. Así el valor representa la altura en un eje blanco-negro, siendo el valor 0 negro y 100 blanco.

7.2. Segmentación por colores

Afortunadamente el color rojo de las señales no es un color muy típico en la vida real. Eso hace que la segmentación por colores sea teóricamente bastante sencilla pero como se verá a continuación surgen numerosos problemas.

La óptica de la cámara no capta los colores como los ve un ojo humano, sino que pierde bastante rango. Otro problema es la iluminación, incluso a las 5 de la tarde cuando hay sobrada iluminación solar depende en gran medida como se ve en la Figura 17 y en la Figura 18 de si el Sol ilumina la señal de frente o desde otro ángulo, ya que la señal pasa de verse con un fuerte color rojo a un rojo bastante oscuro, que en ocasiones llega a ser prácticamente negro. Además en la ciudad se producen numerosas sombras que complican en gran medida la detección. En este ejemplo mientras que a una hora del día las dos señales quedan detectadas, las mismas señales cuatro horas más tarde no aparecen si quiera en la fase de segmentación.

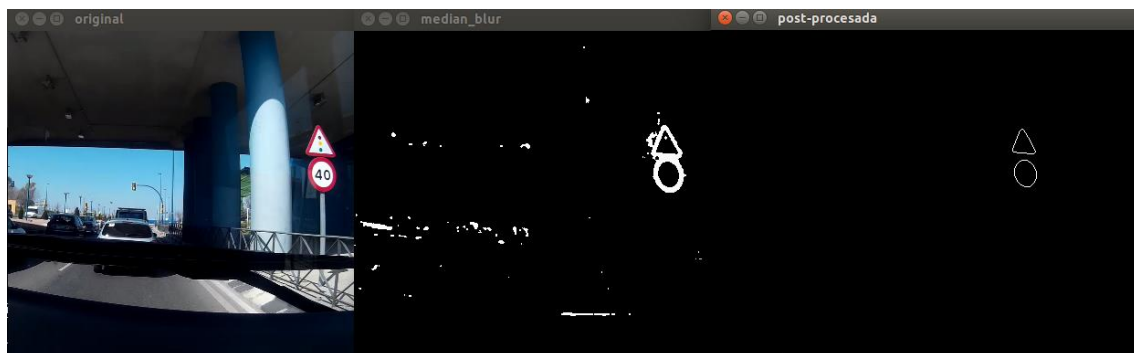


Figura 17 - Señal 'A' a las 14:00

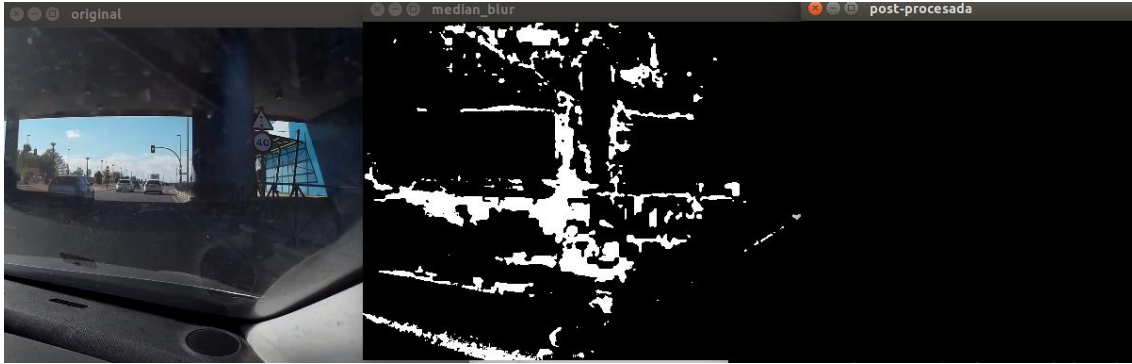


Figura 18 – Señal 'A' a las 18:30

Para tratar de solucionar este problema se disminuyeron los umbrales de la saturación y del valor, esperando así captar señales que no estuvieran bien iluminadas. Así aumenta el rango de detección de señales, no obstante también aumenta el número de falsos positivos, ya que ahora se admiten colores que al empezar el proyecto no se pensaban que pudieran formar parte de una señal roja. Gran parte de la imagen pasa ahora la segmentación por colores, para tratar de solucionar este problema se lleva a cabo una etapa de post-procesamiento.

También se llevó a cabo un intento de detección de señales durante la noche, no obstante los resultados fueron nulos. La carretera en la que se llevó a cabo la prueba estaba poco iluminada, y necesitaba usar las luces de largo alcance, al alcanzar estas luces las señales reflejan tanta luz que la señal se ve casi completamente blanca Figura 19 y no se detecta prácticamente ningún tono rojo. Al aproximarse a la señal, ésta ya no refleja tanta luz, por lo que se pueden apreciar más los tonos rojos, sin embargo no es suficiente para detectar forma alguna que se pueda corresponder con una señal Figura 20. Esto hace que la detección de señales de noche quede fuera del ámbito de aplicación de este proyecto.

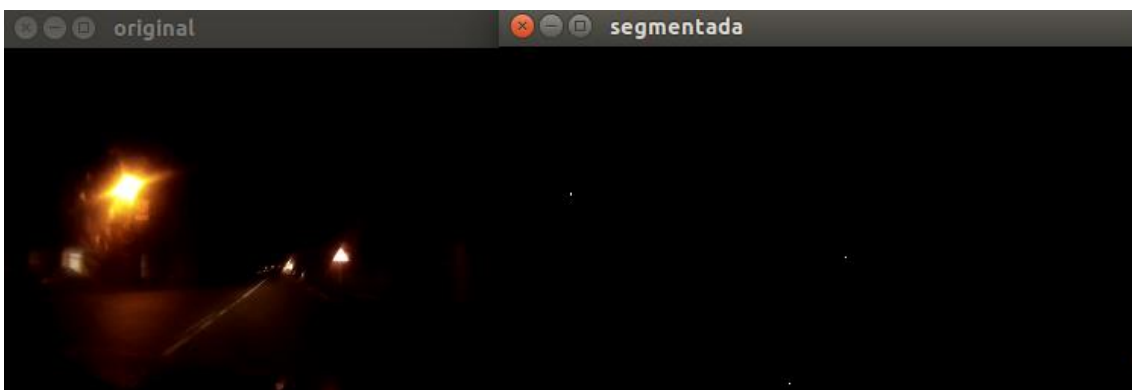


Figura 19 – Señal captada de noche iluminada por las luces de carretera

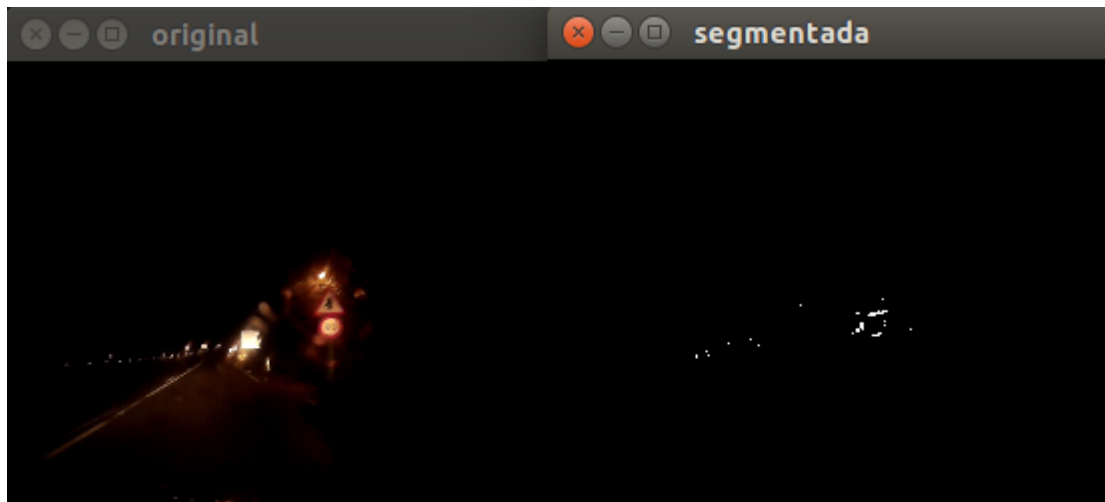


Figura 20 – Señales captadas de noche

También se producen serios problemas de segmentación del color al trabajar a contraluz. Como se puede ver en Figura 21 y en Figura 22, los colores de la señal no guardan relación con los colores esperados y que se han establecido en los valores umbrales. Los valores umbrales no se pueden bajar más ya que provocaría un aumento desproporcionado de los falsos positivos, por lo tanto se decide obviar la detección de señales bajo estas condiciones.

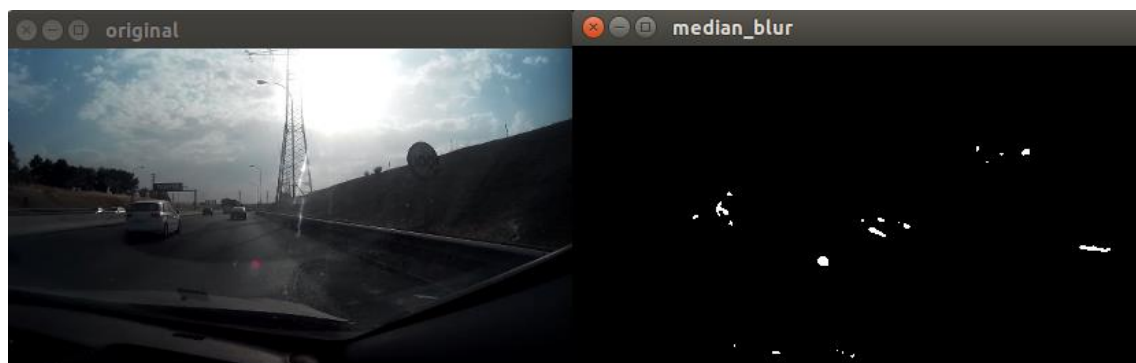


Figura 21 – Señal captada a contraluz

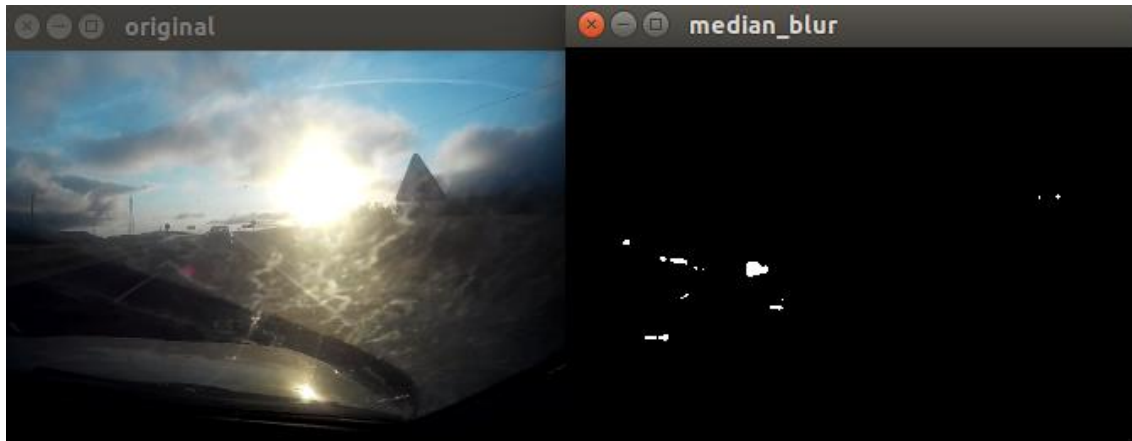


Figura 22 - Señal captada a contraluz

Tras realizar la etapa de segmentación el ordenador vería ahora las imágenes captadas por la cámara como se ven en Figura 23.

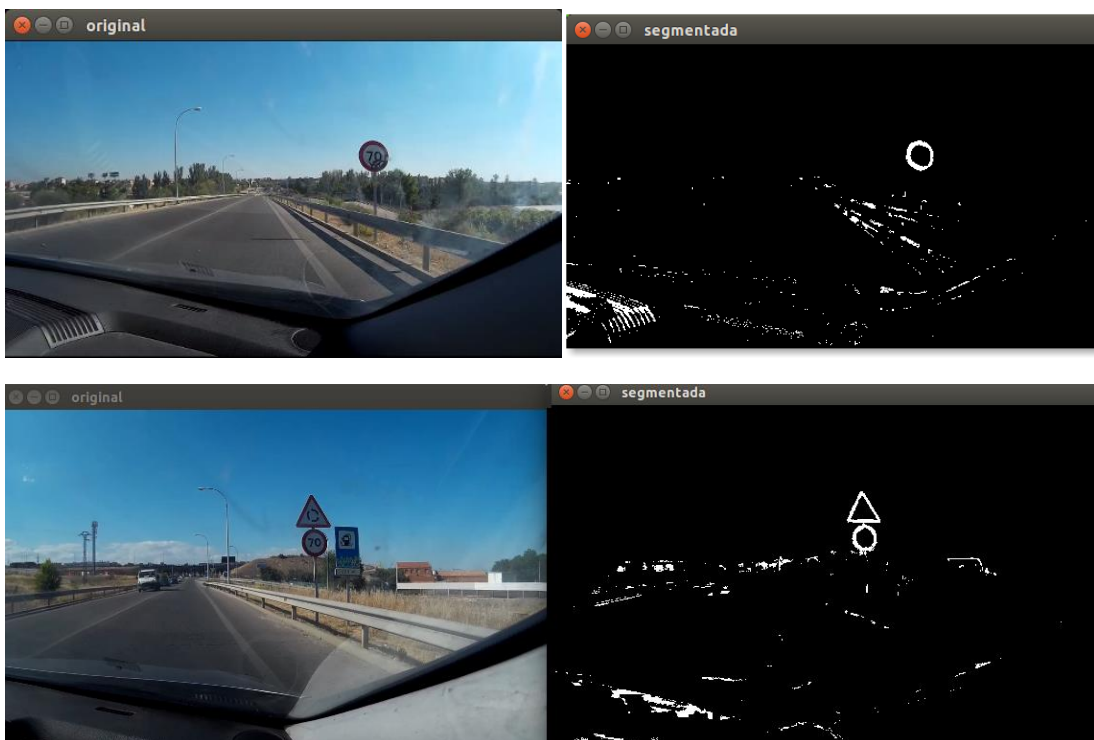


Figura 23 – Etapa de segmentación de varias imágenes

7.3. Post-procesamiento

Como se puede ver en la Figura 23 tras la segmentación por colores aún queda una gran cantidad de la imagen que no guarda relación alguna con las señales que queremos detectar pero que contribuye a aumentar el número de falsos positivos en gran medida. En la Figura 24 si se intenta detectar círculos o triángulos en la imagen simplemente segmentada, se pueden obtener falsos positivos que ni si quiera guardan relación con las formas buscadas.

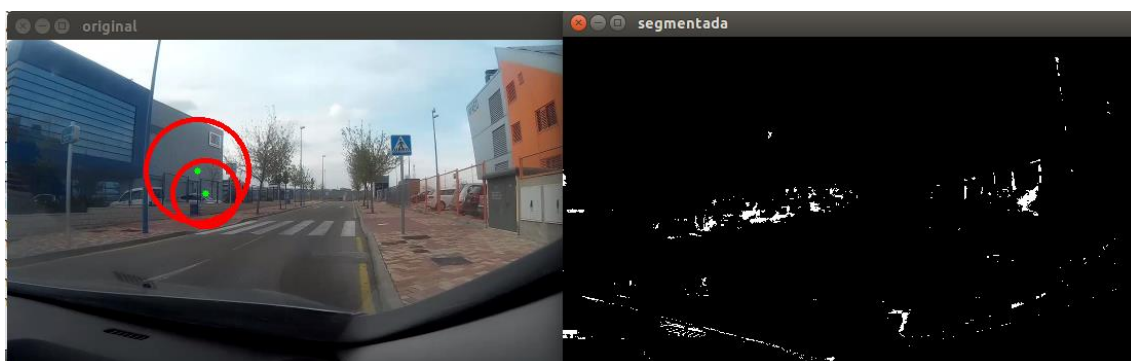


Figura 24 – Detección de círculos en la imagen segmentada

Lo primero que se hace es aplicar un filtro de mediana. El filtro de la mediana funciona ordenando los valores de los píxeles de menor a mayor en torno a uno dado (el número depende del kernel elegido) y aplicando el valor que quede en el medio al píxel objetivo.



Figura 25 – Filtro de mediana 3x3

Una de las ventajas del filtro de la mediana respecto al de la media es que no funciona promediando los valores de los vecinos, sino que le aplica un valor de píxel existente. Esto hace que sea ideal para eliminar el ruido impulsional que aparece en ocasiones. Además el filtro de la mediana suaviza las formas y evita muchos falsos positivos que surgían en la detección de círculos.



Figura 26 – Resultado de aplicar el filtro de la mediana

A continuación se procede a calcular todos los contornos de la imagen filtrada y sin establecer una jerarquía. Estos contornos pasan por dos filtros más. El primero de tamaño, se descartan las formas muy pequeñas o excesivamente grandes que no podrían corresponderse con señales de tráfico. El segundo funciona calculando el *aspect ratio* de las *bounding box* que encierran los contornos restantes. La bounding box de una señal de tráfico debería tener teóricamente un valor de 1, ya que el valor del rectángulo sin girar más pequeño que encierra un círculo o un triángulo equilátero como los de las señales es siempre un cuadrado. Dando cierto margen para considerar que la señal puede estar girada, y por tanto aparecer deformada, los márgenes se establecen entre 0,75 y 1,15, ya que lo usual es que en caso de estar deformada sea más alta que ancha ($\text{aspect_ratio} = \frac{\text{width}}{\text{height}}$).



Figura 27 – Señales captadas giradas por la proximidad de la cámara.

Finalmente a los contornos restantes en la imagen se les aplica la envoltura convexa. Con esta operación obtenemos los menores conjuntos de puntos que contienen a los contornos restantes, esto permite corregir problemas en la detección de la señal debido a que el contorno no esté completo, bien sea por el estado de la señal (en ocasiones hay pintadas o pegatinas adheridas) o simplemente porque el color no es el ideal, no se detecta la forma completa. La envoltura convexa permite reconstruir “un agujero” en un contorno, si bien si hubiera dos se considerarían dos contornos distintos y no se podría solucionar.

Estas dos condiciones disminuyen drásticamente el número de falsos positivos generados en la detección de señales provocado por los aumentos de rango de colores, sin embargo, a veces también puede causar errores, puesto que a veces 2 señales que están una encima de otra no las ve como dos distintas sino como un único contorno y produce errores ya que no pasan el filtro del *aspect ratio* al tener el doble o triple de altura de que de anchura, Figura 28. En la Figura 29 vemos el mismo tipo de fallo, aquí el triángulo no estaría completo y

probablemente no sería detectado en la fase posterior, sin embargo impide que detecte también el círculo.

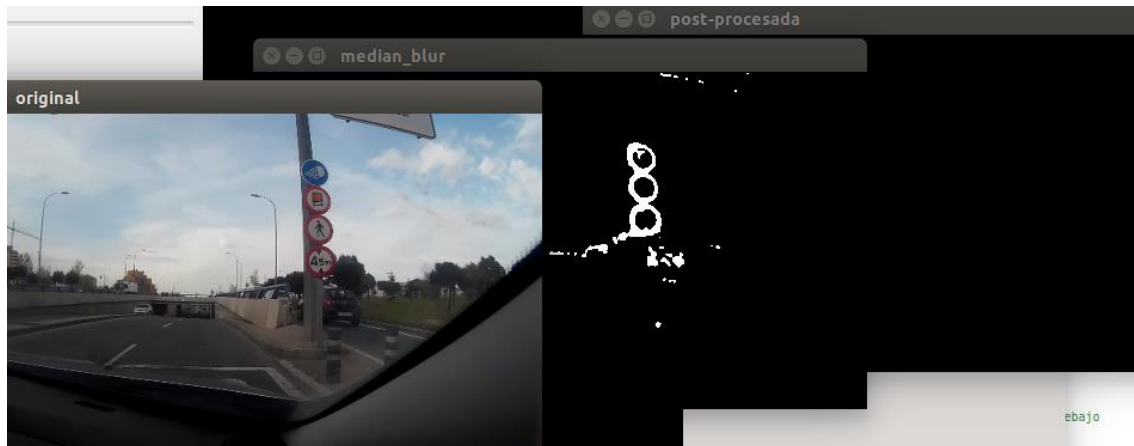


Figura 28 – Fallo en la detección de contornos, 3 señales juntas

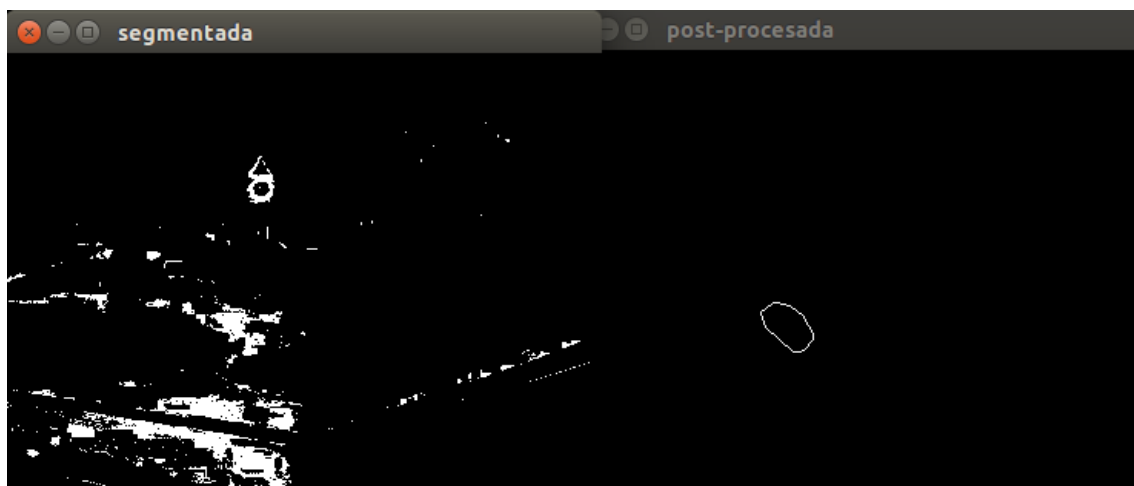


Figura 29 - Fallo en la detección de contornos, 2 señales juntas

No obstante estos fallos relacionados con el aspect ratio influyen en frames aislados, si las señales se captan en varios frames (es lo habitual en condiciones lumínicas aceptables) suelen pasar el filtro (como se puede apreciar en Figura 28 y Figura 31, son frames muy cercanos, y aunque en uno falle en otro acierta y las señales quedan detectadas). En Figura 30 se pueden observar las etapas de segmentación y post procesamiento, se puede ver como finalmente en este caso solo quedan los contornos que se corresponderían con las tres señales que se ven en la imagen.

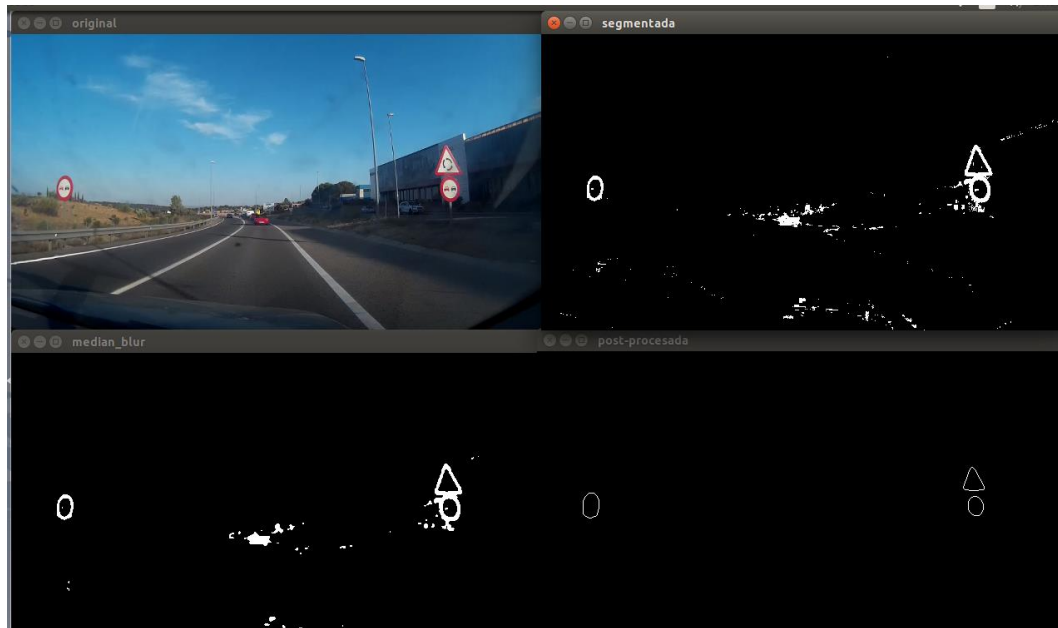


Figura 30 – Proceso completo de segmentación y post-procesamiento

7.4. Detección de círculos y triángulos

Tras realizar el post-procesamiento en la imagen a analizar solo quedan los contornos que potencialmente pueden tratarse de un círculo o de un triángulo.

Para llevar a cabo la detección de estas formas se utilizarán las funciones pertenecientes a las librerías de las OpenCV: *HoughCircles* y *approxPolyDP*. Esta será la última fase de detección de las señales, los contornos que sean identificados como círculos como triángulos pasarán a ser considerados como señales de prohibido o precaución.

HoughCircles se usará para detectar los círculos. Funciona de la siguiente manera. La función recibe una imagen en escala de gris, y realiza un cálculo de los bordes mediante el detector de Canny. A continuación se produce el proceso de detección de círculos, que es relativamente sencillo:

- Primero se crea un espacio acumulador, que está compuesto de una celda para cada píxel, inicializadas todas a 0.
- Para cada punto de un borde de la imagen (i, j) se incrementan todas las celdas que de acuerdo con la ecuación de un círculo $((i-a)^2 + (j-b)^2 = r^2)$ podrían ser el centro de un círculo, todas estas celdas están representadas por la letra a en la ecuación.
- Para todos los valores posibles encontrados en el paso anterior se encuentran todos los posibles valores de b que satisfacen la ecuación.

- Se buscan los máximos locales de las celdas, estos son todas las celdas cuyo valor es mayor que el resto de sus celdas vecinas. Estas celdas son las que tienen una mayor probabilidad de ser la localización de los círculos que se tratan de encontrar.

Esta función localiza con bastante precisión los centros de los círculos, sin embargo en ocasiones tiene más problemas a la hora de calcular el radio exacto, es por esto que a la hora de seleccionar la ROI se amplía el marco, para asegurarse de que se capta la señal completa.

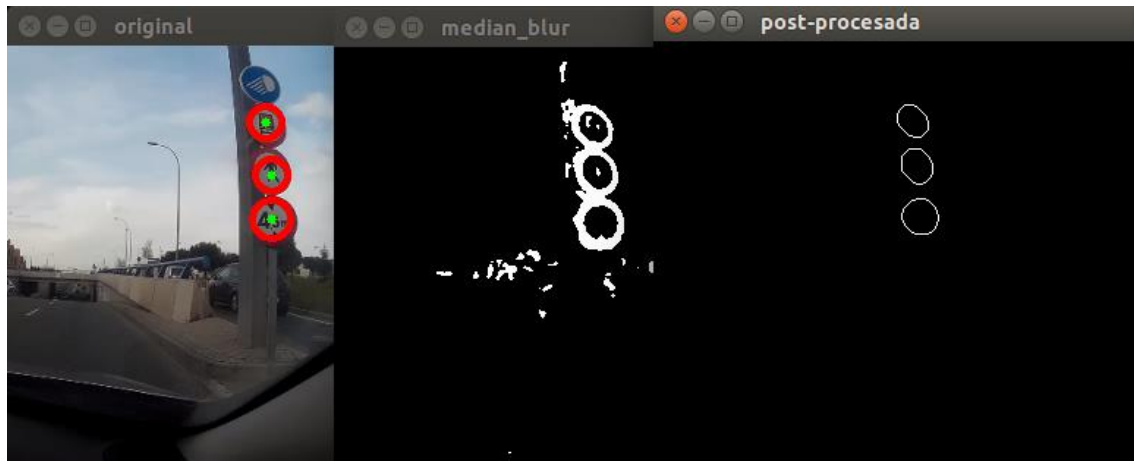


Figura 31 – Detección de círculos en la imagen

Para la detección de triángulos se utiliza la función *approxPolyDP*. Esta función lo que hace es aproximar los contornos que se le introducen en polígonos o curvas, con menos vértices y siempre que la distancia entre el polígono calculado y el contorno original sea menor que la precisión especificada.

Una vez aproximados los contornos a polígonos se seleccionan aquellos que tengan tan solo 3 lados. A continuación se comprueban las longitudes de los lados, puesto que es un triángulo equilátero no debería haber gran diferencia entre los lados (aun considerando el giro con el que se hayan podido captar las señales), por lo tanto sólo se aceptan aquellos triángulos que cumplan que el lado menor tiene como mínimo el 65% de longitud del lado mayor.



Figura 32 – Detección de triángulos y extracción de la ROI.

Hasta aquí la fase de detección de señales estaría completa, se detectan con bastante acierto tanto señales triangulares como circulares, aunque como se ha comentado antes la tasa de acierto depende mucho de la iluminación de las señales.

8. Reconocimiento de señales

El reconocimiento de señales se lleva a cabo mediante un clasificador de cascada de características tipo Haar. Para hacerlo funcionar se han tenido que recoger muestras de las señales con las que entrenar un clasificador distinto para cada señal a detectar, y finalmente se pasa cada clasificador por cada triángulo o círculo detectado para comprobar si la señal detectada está en la base de datos.

8.1. Clasificador de cascada de características tipo Haar

El reconocimiento de las señales de tráfico se basará en [11] "Haar Feature-based Cascade Classifier for Object Detection". Un clasificador se entrena a partir de cientos de muestras de la imagen que se quiera reconocer, y también con imágenes negativas que se utilizarán como fondo, y que no tienen por que guardar relación alguna con la imagen que se quiera detectar.

Después de que un clasificador esté entrenado se le puede aplicar a la región de interés de nuestra imagen donde hemos detectado una posible señal. El clasificador devuelve un 1 si la detección es positiva y un 0 si es al contrario, por lo que necesita entrenar un clasificador para cada tipo de señal que se quiera reconocer. Se tendrán dos etapas de reconocimiento, la de las señales triangulares que actuará donde se haya detectado un triángulo y la de las señales circulares, que hará lo propio donde se haya localizado un círculo. Los clasificadores están diseñados para que puedan ir variando el tamaño para así ser capaces de reconocer la señal si está más cerca o más lejos.

El nombre de cascada significa que el clasificador está compuesto de diferentes clasificadores más pequeños o etapas, que se van aplicando en la región de interés hasta que se rechace o se pasen todas las etapas y se considere el objeto detectado. Los clasificadores básicos son del tipo de árbol de decisión con al menos 2 "hojas". Las características de tipo Haar son las entradas a los clasificadores básicos, el algoritmo actual usa las siguientes:

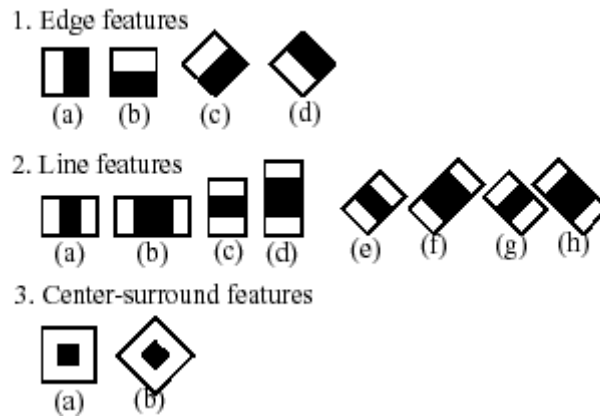


Figura 33 – Características tipo Haar [11]

Las características usadas en un clasificador están especificadas por su forma (1a, 2b, etc.), posición dentro de la ROI y su escala. Por ejemplo en el caso de la característica de la tercera línea (2c) la respuesta es calculada como la diferencia entre la suma de los píxeles bajo el rectángulo que cubre la característica completa (incluyendo las dos líneas blancas y la negra del medio) y la suma de los píxeles bajo la línea negra multiplicados por 3 para compensar la diferencia de áreas, en función de ese resultado se decide si esa característica cumple los requisitos o no.

Las primeras etapas suelen tener muy pocas características para poder descartar rápidamente si no hay nada que se asemeje al objeto buscado. Por ejemplo en el clasificador detector de caras hay 38 etapas, las 5 primeras tienen 1 característica, 10, 25, 25, 50, y en total hay unas 6000 características. Los clasificadores entrenados en este proyecto suelen tener entre 8 y 12 etapas, depende del número de imágenes que se usen y la dificultad que encuentre el programa en cumplir los requisitos especificados (para todas las señales se han usado los mismos)

8.2. Preparación de las muestras

8.2.1. Muestras positivas

Para obtener las muestras positivas se utiliza el programa de detección de imágenes. Se procesan más de 73 archivos de video con alrededor de 100 minutos de grabación y se extraen de ellos todas las señales que el programa es capaz de detectar. Algunos se pasan más de una vez por el detector para obtener más señales ya que la detección no siempre es la misma, puesto que los frames que se analizan varían. Toda señal detectada es guardada para su posterior clasificación.

A continuación se procede a clasificar las señales por tipo, aun habiendo procesado una gran cantidad de videos y hecho múltiples pasadas, a la hora de agrupar las señales en carpetas hay muchos tipos de señales de las que casi no se tienen muestras, especialmente en cuanto a las señales de precaución, por ejemplo de las señales de precaución semáforo, peatón, animales salvajes u otras como velocidad máxima 120 km/h se tienen muy pocas. Mientras que de otras como ceda el paso, 70 km/h o 40 km/h se han obtenido un gran número que permite entrenar mejor a los clasificadores.

Una vez agrupadas las muestras se procede a recortar la región de interés dentro de la propia señal y a igualar los tamaños. Se consideró la posibilidad de aumentar el número de muestras tratando las imágenes obtenidas de los vídeos mediante variaciones de la iluminación, introducción de ruidos, rotaciones de la imagen.... o recurrir a bases de datos de internet. Sin embargo de acuerdo a [12] se obtienen mejores resultados con imágenes del mundo real que con aquellas generadas artificialmente, además se decidió usar sólo imágenes capturadas por la SJ4000 para aumentar la tasa de éxito del reconocimiento en lugar de recurrir a bases de datos.



Ilustración 5 – Imagen captada por la SJ400

Ilustración 4 – Imagen de entrenamiento

8.2.2. Muestras negativas

Para las muestras negativas se recurre a una base de datos de internet localizada en [13] donde se descargan más de 1000 muestras, aunque a la hora de realizar el entrenamiento no se usen tantas, se decide usar el doble de negativas que de imágenes positivas.

Estas imágenes como se dijo antes no tienen que guardar ningún relación particular con el objeto a detectar, simplemente se usarán como imágenes de fondo sobre las que se situarán las imágenes positivas a la hora del entrenamiento.



Ilustración 6 – muestra de las imágenes negativas usadas en el entrenamiento

8.3. Entrenamiento

El entrenamiento se realiza de acuerdo a [14]. Una vez que se tienen las imágenes recortadas para que tan solo se vea el elemento a reconocer, se crea un fichero (*samplesdescription.dat*) con las rutas de las imágenes, el número de señales en cada imagen y la localización del ROI donde se encuentra la señal dentro de la imagen (en este caso coincide con las dimensiones de la imagen puesto que está recortada a tal propósito).

```
./230.jpg 1 0 0 28 28  
./670.jpg 1 0 0 28 28  
./330.jpg 1 0 0 28 28  
./490.jpg 1 0 0 28 28  
./280.jpg 1 0 0 28 28  
./770.jpg 1 0 0 28 28  
./160.jpg 1 0 0 28 28
```

Figura 34 – Extracto de samplesdescription.dat

A continuación se crea el archivo que contendrá todas las imágenes de entrenamiento y que se usará para entrenar el clasificador mediante la función de OpenCV *createsamples*.

```

daniel@VanDamme:~/Escritorio/Entrenamiento/Imagenes/ceda_2$ opencv_createsamples
-info samplesdescription.dat -vec samples.vec -w 35 -h 35
Info file name: samplesdescription.dat
Img file name: (NULL)
Vec file name: samples.vec
BG file name: (NULL)
Num: 1000
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Width: 35
Height: 35
Create training samples from images collection...
samplesdescription.dat(321) : parse errorDone. Created 320 samples

```

Figura 35 – Creación de samples.vec

Una vez obtenido el archivo *samples.vec* se puede comenzar el entrenamiento del clasificador. En las Figura 36 y Figura 37 se pueden ver las etapas finales del entrenamiento del clasificador. La duración depende mucho del número de imágenes positivas, como se ve Figura 36 tarda poco más de 40 minutos ya que tenía unas 40 muestras positivas, sin embargo en Figura 37 se ve que requiere más de 6 horas de entrenamiento al contar con más de 320 imágenes de señales. En un entrenamiento realizado con unas 1000 imágenes positivas (obtenidas mediante variaciones de las de la base de datos, como introducción de ruido gaussiano o rotaciones) llegó a tardar más de 30 horas, sin embargo parece que con unas 100 imágenes de entrenamiento se pueden obtener resultados bastante buenos.

```

===== TRAINING 13-stage =====
<BEGIN
POS count : consumed    89 : 89
NEG count : acceptanceRatio    190 : 9.79037e-07
Precalculation time: 2
+---+-----+-----+
| N |   HR   |   FA   |
+---+-----+-----+
|  1 |       1 |       1 |
+---+-----+-----+
|  2 |       1 | 0.642105 |
+---+-----+-----+
|  3 |       1 | 0.536842 |
+---+-----+-----+
|  4 |       1 | 0.326316 |
+---+-----+-----+
END>
Training until now has taken 0 days 0 hours 40 minutes 36 seconds.

===== TRAINING 14-stage =====
<BEGIN
POS count : consumed    89 : 89
NEG count : acceptanceRatio    190 : 4.17803e-07
Required leaf false alarm rate achieved. Branch training terminated.

```

Figura 36 – Entrenamiento del clasificador de prohibido adelantar

```

===== TRAINING 16-stage =====
<BEGIN
POS count : consumed    320 : 320
NEG count : acceptanceRatio    640 : 1.4547e-06
Precalculation time: 22
+-----+-----+
|  N  |    HR   |    FA   |
+-----+-----+
|  1  |        1|        1|
+-----+-----+
|  2  |        1|        1|
+-----+-----+
|  3  |        1|        1|
+-----+-----+
|  4  |        1| 0.934375|
+-----+-----+
|  5  |        1| 0.8375|
+-----+-----+
|  6  |        1| 0.71875|
+-----+-----+
|  7  |        1| 0.709375|
+-----+-----+
|  8  |        1| 0.560938|
+-----+-----+
|  9  |        1| 0.451562|
+-----+-----+
END>
Training until now has taken 0 days 6 hours 4 minutes 5 seconds.

===== TRAINING 17-stage =====
<BEGIN
POS count : consumed    320 : 320
NEG count : acceptanceRatio    640 : 8.08717e-07
Required leaf false alarm rate achieved. Branch training terminated.

```

Figura 37 – Entrenamiento del clasificador de ceda el paso

Hasta el momento y por falta de una base de datos con más señales se han entrenado clasificadores para las siguientes señales:

- Prohibido superar 100 km/h
- Prohibido superar 90 km/h
- Prohibido superar 70 km/h
- Prohibido superar 60 km/h
- Prohibido superar 40 km/h
- Prohibido adelantar
- Prohibida la circulación a camiones
- Ceda el paso
- Precaución glorieta

8.4. Reconocimiento

La Figura 38 muestra la estructura de la fase de reconocimiento de señales circulares, para las señales triangulares sería igual pero con los clasificadores correspondientes. Cada ROI que se ha detectado es pasado por todos los clasificadores para tratar de reconocer la señal, el problema de este método es que a veces varios clasificadores pueden reconocer la señal como si se tratara de la suya, ya que el funcionamiento entre ellos es independiente, no es como por ejemplo en una red neuronal que da solamente el resultado que más se aproxima a la señal analizada. En caso de que el clasificador considere que la señal que busca se encuentra en la ROI manda un mensaje a la terminal, pero independientemente del resultado actúan los siguientes clasificadores en la misma ROI.

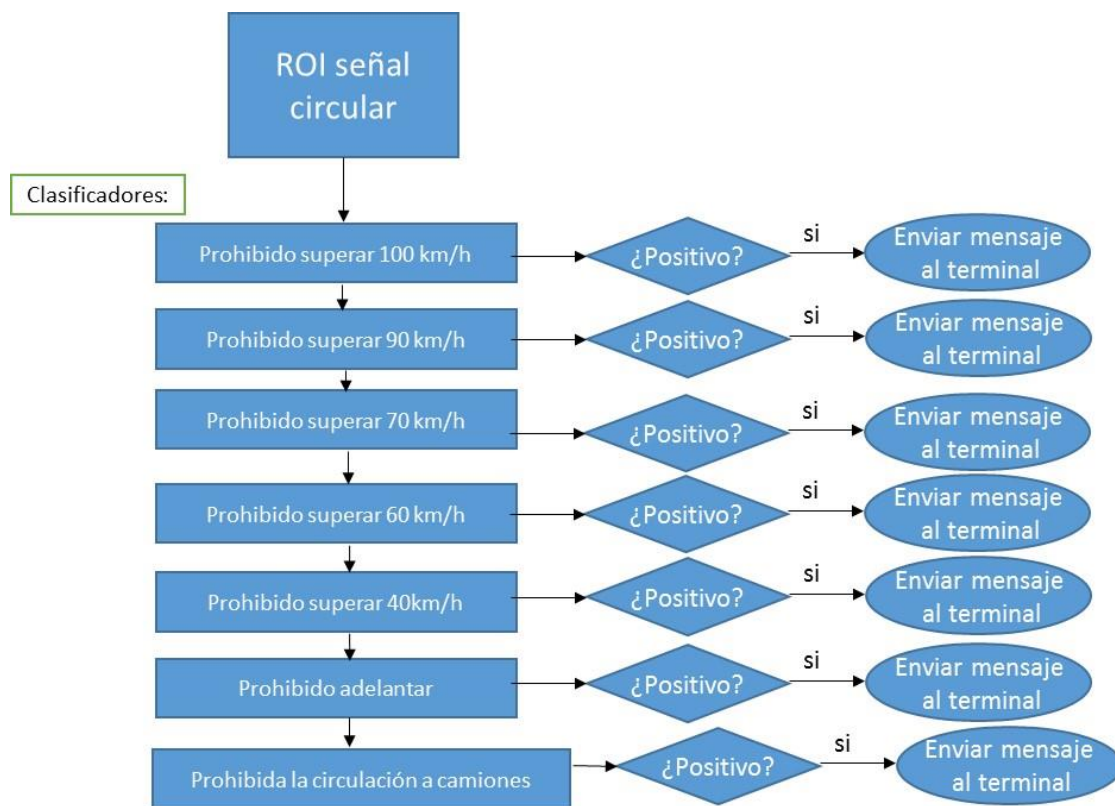


Figura 38 - Estructura de la fase de reconocimiento de las señales circulares

En las figuras Figura 39 Figura 40 Figura 41 se puede observar el funcionamiento final del programa en el que se reconocen distintas señales, ceda el paso, prohibido adelantar..... En la Figura 42 el programa reconoce a la vez dos señales distintas como son las de precaución glorieta y prohibido superar los 70 km/h, el funcionamiento del programa permite reconocer varias señales en la misma imagen sin problemas adicionales si se han entrenado los clasificadores correspondientes.

También se puede ver en la Figura 43 cómo el clasificador de 60 km/h produce un falso positivo al analizar una señal de 40 km/h. Como se discutirá en Resultados los clasificadores de 90 y de 60 km/h producen un número especialmente alto de falsos positivos, por lo que se descarta su uso.



Figura 39 – Reconocimiento de una señal de ceda el paso



Figura 40 - Reconocimiento de una señal de prohibido adelantar

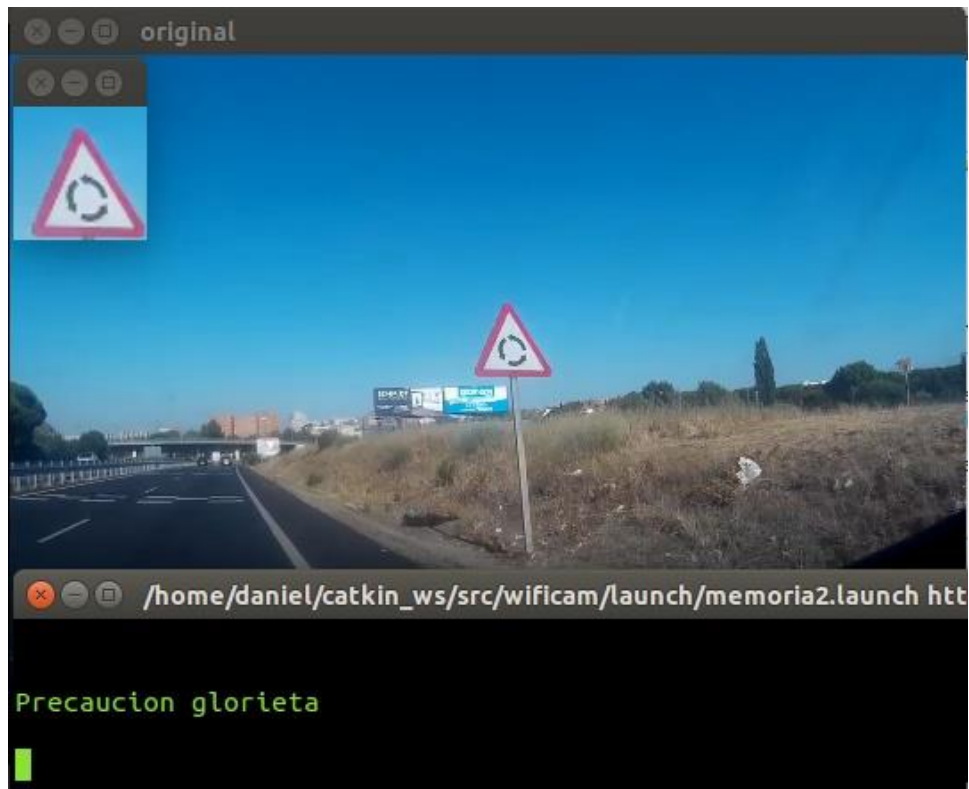


Figura 41 - Reconocimiento de una señal de aviso de una glorieta



Figura 42 - Reconocimiento de una señal de 70 km/h y una de glorieta en la misma imagen



Figura 43 - Reconocimiento de una señal de prohibido superar los 40 km/h y un falso positivo de 60 km/h

9. Resultados

9.1. Tiempo de ejecución

Los resultados de tiempo de ejecución del programa final se han realizado sobre los videos grabados a una calidad de 1280x720. Como se observa en Figura 44 el primer nodo *wificampc* envía muchas más imágenes de las que el nodo de detección (tiempo de ejecución) es capaz de procesar. El tiempo total de ejecución depende como veremos luego si en la imagen hay más elementos que puedan confundirse con círculos o menos, pero suele estar en torno a los 100-110 ms, lo que nos daría una velocidad de procesamiento de casi 10 frames/s, lo que es una tasa aceptable, situándose en la frontera del tiempo real.

```
tiempo de ejecucion: 106.875 milisegundos  
  
tiempo de envio: 15.6988 milisegundos  
  
tiempo de envio: 9.38313 milisegundos  
  
tiempo de envio: 8.05391 milisegundos  
  
tiempo de envio: 5.88678 milisegundos  
  
tiempo de envio: 6.90749 milisegundos  
  
tiempo de ejecucion: 101.874 milisegundos  
  
tiempo de envio: 8.71533 milisegundos
```

Figura 44 – Tiempo de cómputo total

Introduciéndose un poco más en el funcionamiento interno del programa podemos ver como la parte que más tiempo consume es la de detección de círculos, que es del orden de 100 veces mayor que la de triángulos. Esto se debe a que el algoritmo de detección de círculos *HoughCircles* que proporciona OpenCV exige mucho tiempo de cómputo. Podemos ver variaciones considerables en el tiempo de detección en distintas imágenes, esto se debe a que dependiendo del frame puede haber más formas que se asemejen a un círculo o a un triángulo y que por tanto necesite más tiempo el programa para verificarlo.

```

Deteccion de circulos  92.0531  milisegundos
Deteccion de triangulos>  4.35623  milisegundos
Deteccion de circulos  83.5519  milisegundos
Deteccion de triangulos>  1.77291  milisegundos
Deteccion de circulos  108.302  milisegundos
Deteccion de triangulos>  1.6797  milisegundos
Deteccion de circulos  137.149  milisegundos
Deteccion de triangulos>  5.59051  milisegundos
    
```

Figura 45 – Tiempo de detección y reconocimiento

Para la evaluación del programa final se han usado videos grabados con señales donde la visibilidad de las mismas es buena. Se ha decidido prescindir de los videos donde hay muchas sombras ya sea por poca iluminación, por contraluz o por edificios. No obstante en los videos donde la visibilidad media a lo largo de la grabación es buena se han tenido en cuenta todas las señales, independientemente de si el coche iba más o menos rápido, o iba por el carril central y las señales al estar en los extremos se ven peor. Las señales que están en un carril adyacente pero que no afectan a nuestro coche (carriles de aceleración, salidas....) no se han tenido en cuenta para la detección, aunque si para el reconocimiento (ya que no hay necesidad de detectarlas en la realidad pero una vez detectadas se pueden usar para evaluar el clasificador).

9.2. Detección

En lo que respecta a la detección de las señales, durante los videos analizados había un total de 441 señales factibles de detectar, de ellas 354 fueron detectadas correctamente, 87 fueron pasadas por alto, y se produjeron 7 falsos positivos. Donde más falsos positivos se producen es entorno urbano, aunque la mayor parte de los videos en ciudad no están reflejados en este análisis puesto que había muy pocas señales y el índice de acierto era muy bajo; sin embargo los edificios a menudo tienen un color que está dentro de los rangos de segmentación y en estas zonas aparece una mayor concentración de falsos positivos.

falsos negativos	87	19,73%
verdaderos positivo	354	80,27%
falsos positivos	7	1,56%
Totales	441	100%

Tabla 1 – Resultados finales de la detección

Como se aprecia el porcentaje de falsos positivos es bastante bajo (sólo un 1,56% de los resultados totales). Esto podría incitar a aumentar los parámetros de búsqueda de las señales para tener un rango mayor aún a riesgo de tener un mayor número de falsos positivos, no obstante esto ya se intentó y los resultados no eran muy buenos, puesto que a menudo el problema era el color de las señales, y aumentando el rango no se conseguía mejorar la detección porque al haber una mayor cantidad de información se solapaba la útil con el fondo y la detección no mejoraba. Por lo tanto el aumentar el rango de búsqueda no es tan directo como podría parecer.

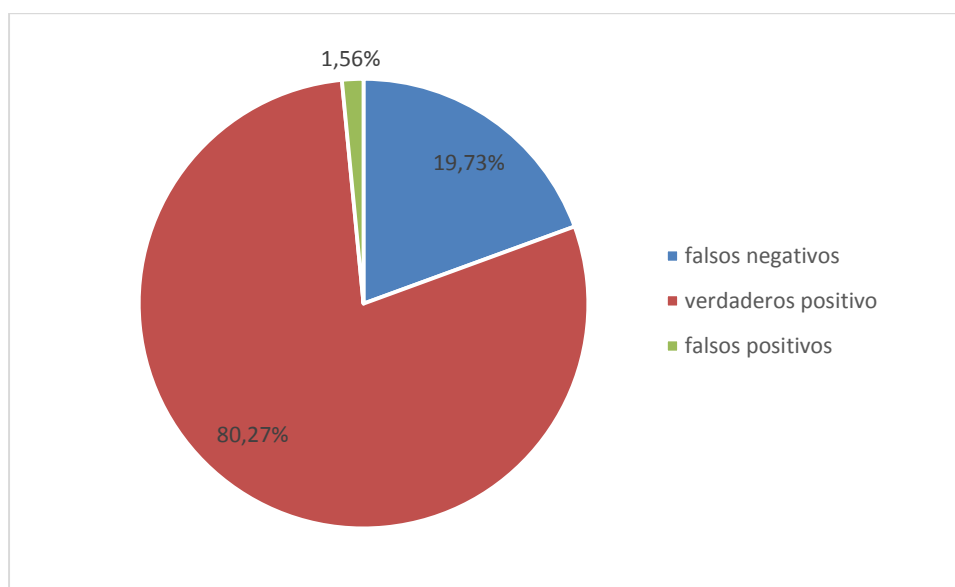


Figura 46 – Resultados finales de la detección

9.3. Reconocimiento

En cuanto al reconocimiento de las señales detectadas, de las 242 señales que podían ser reconocidas, fueron clasificadas con éxito 215 de ellas. El problema viene más con los falsos positivos, donde se produjeron 74. La mayor parte de ellos (83%) está concentrado en dos clasificadores, el de 60 km/h y el de 90 km/h, como se comentó antes estos clasificadores a pesar de haber seguido el mismo proceso de entrenamiento y con un número similar (incluso mayor en el caso de la de 60 km/h) que el resto de clasificadores producen un mayor número de errores, probablemente se trate simplemente de la similitud de las formas, que se confunde más fácilmente con el resto de señales.

De todos modos hay que mencionar que está marcado como falso positivo aunque sólo interprete erróneamente un frame en el que aparece la señal. Lo normal durante el

reconocimiento es que en varias imágenes del video se reconozca la señal, y el falso positivo a veces se da sólo en algunas de ellas.

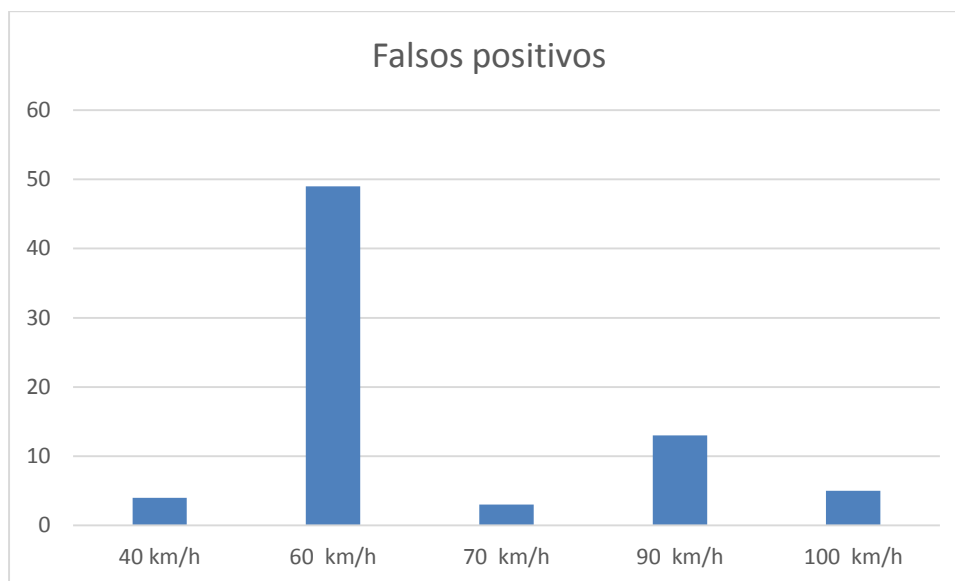


Figura 47 – Número total de falsos positivos encontrados por clasificador

Las señales triangulares, y las de prohibido adelantar y prohibida la circulación de camiones no dan un solo falso positivo. Esto se debe a que estas señales son bastante distintas del resto, no es como el caso de los números en las de velocidad máxima. Sin embargo si se entrenaran clasificadores de tipo cascada de las señales de precaución intersección por la izquierda, derecha, izquierda y derecha a la vez, y las de incorporación, lo más probable es que dieran un gran número de falsos positivos entre ellas, puesto que son muy similares unas a otras, pero no se ha dispuesto de imágenes de entrenamiento suficientes para entrenar estos clasificadores.

Como se puede observar el porcentaje de acierto en el reconocimiento de señales es bastante alto (teniendo en cuenta la poca cantidad de muestras de la que se dispone en muchos casos). El menor porcentaje de acierto estaría en las señales de ceda el paso (76%), esto se debe a que el entrenamiento necesitaría más señales, puesto que en este caso el ROI de entrenamiento tiene que incluir el triángulo completo y por tanto el fondo incluido en el cuadrado, por lo que hay una mayor variabilidad que si sólo se analiza el interior de la señal como en el resto de casos.

El número de muestra a analizar en los casos de 90 km/h, 100 km/h, y prohibido para camiones es bastante bajo (8,10 y 9 respectivamente), por lo que los resultados no son tan fiables como por ejemplo en el caso de 40 km/h (49 muestras), sin embargo tienen una tasa de acierto similar al resto, lo que incita a pensar que los resultados no están muy desencaminados de la realidad. La tasa media de acierto es del 88%, lo que es buen resultado, más teniendo en cuenta que normalmente los entrenamientos de los clasificadores suelen entrenarse con muchas más muestras de las que se disponía en la base de datos.

Del total de 242 señales a reconocer, los clasificadores producen un total de 74 falsos positivos, que sería un número inasumible. Sin embargo si se tiene en cuenta que 62 de ellos han sido en las señales de 60 y 90 km/h que como se comentó quedarán excluidos del funcionamiento final del programa, quedaría un resultado de 12 falsos positivos en 242 señales, que sería un 5%, esto es un resultado mucho mejor y que se podría considerar como válido, más teniendo en consideración que siempre se detecta en más frames la señal correcta que en los que se produce el falso positivo.

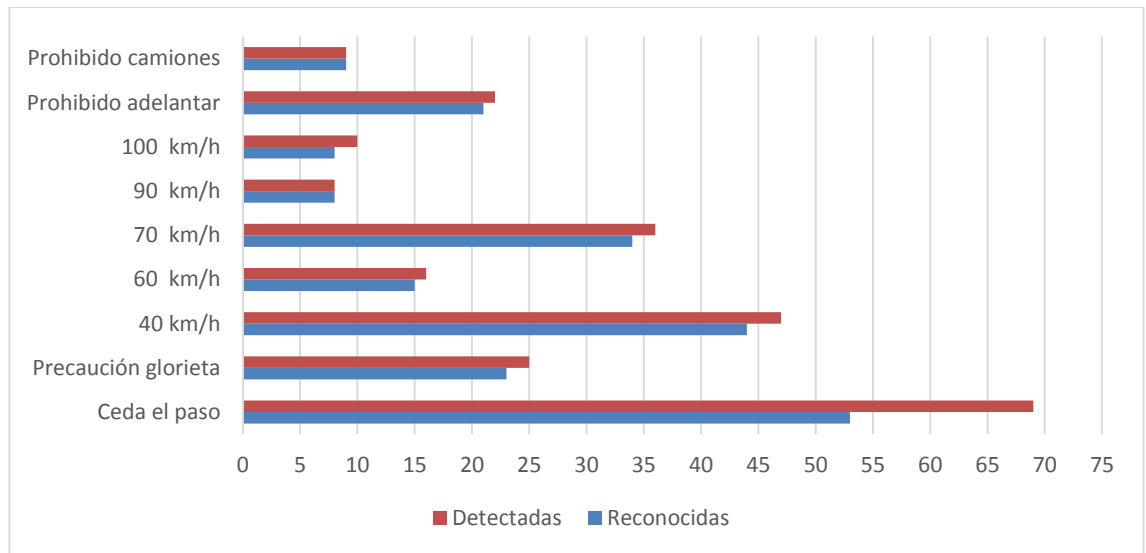


Figura 48 – Resultados del reconocimiento de señales

Señales	Reconocidas	Falsos positivos	Reconocidas (%)	Falsos positivos (%)	Detectadas
Ceda el paso	53	0	76,81%	0,00%	69
Precaución glorieta	23	0	92,00%	0,00%	25
40 km/h	44	4	93,62%	5,41%	47
60 km/h	15	49	93,75%	66,22%	16
70 km/h	34	3	94,44%	4,05%	36
90 km/h	8	13	100,00%	17,57%	8
100 km/h	8	5	80,00%	6,76%	10
Prohibido adelantar	21	0	95,45%	0,00%	22
Prohibido camiones	9	0	100,00%	0,00%	9
Total	215	74	88,84%	100,00%	242

Figura 49 – Resultados del reconocimiento de señales

El problema de este tipo de clasificador que requiere uno por cada tipo de señal que se quiera reconocer, es que en señales que se parezcan bastante entre sí, es probable que varios de ellos indiquen que su señal es la correcta. Este error es difícil de solucionar ya que actualmente los parámetros de fiabilidad requerida presentados ya son bastante altos, sin embargo no son suficientes como se ha podido comprobar para evitar los falsos positivos, y aumentando el número de imágenes de entrenamiento lo más probable es que se aumente el número de señales reconocidas correctamente únicamente.

10. Presupuesto

De acuerdo con las nuevas tablas de amortizaciones publicadas por el Ministerio de Hacienda este año 2015 [15], los equipos electrónicos tienen una amortización del 20%, los equipos de video del 33% y la de los medios de transporte externo es del 16%.

El desarrollo del proyecto ha llevado unos 7 meses de trabajo invirtiendo 4 horas diarias, por tanto ese es el tiempo que se usará para calcular las desamortizaciones y el sueldo del desarrollador.

Para calcular la amortización de los elementos usados durante el proyecto se tendrá en cuenta un % del uso estimado que se les ha dado durante el tiempo de desarrollo del proyecto, puesto que el coche ha sido mucho menos usado que el ordenador por ejemplo. Dichos elementos son los siguientes:

$$\text{Precio de desamortización} = \text{Precio} * \frac{\text{uso}}{100} * \text{coef. amortización} * \frac{7 \text{ meses}}{12 \text{ meses}}$$

Elemento	Precio (€)	Uso	Coficiente de amortización	Precio de desamortización (€)
SJ400	98.99	20%	0.33	3.81
Ordenador	766 .50	90%	0.2	80.43
Coche	12.500	10%	0.16	116.66

Tabla 2 – Costes de material

Durante las sesiones de grabación de tramos de carretera con señales se estima que han invertido unas 12 horas en el coche, suponiendo un gasto medio de 5 l/h y un precio de la gasolina de 1.15 €. El presupuesto invertido en combustible es de: 69€

Para el coste de personal se tiene en cuenta que según la encuesta de salarios del 2014-2015 para ingenieros industriales, el salario medio para un ingeniero con menos de un año de experiencia trabajando a jornada completa en el sector de la investigación es de 18.119€ brutos al año.

$$\text{Coste} = \frac{18.119 * \frac{4 \text{ horas}}{8 \text{ horas}}}{12 \text{ meses}} = 755 \frac{\text{€}}{\text{mes}} \text{ jornada parcial}$$

Apellidos, Nombre	Categoría	Tiempo dedicado (meses)	Coste (€/mes)	Coste total (€)
Cruzado Hernando, Daniel	Ingeniero	7	755	5284

Tabla 3 – Costes de personal

Por tanto los costes totales del proyecto son:

Elemento	Coste (€)
Material	200.9
Gasolina	69
Personal	5284
Total	5553.9

Tabla 4 – Costes totales

11. Trabajos futuros

A lo largo del desarrollo de este Proyecto han ido apareciendo distintos problemas, algunos de los cuales se han podido solucionar mientras que otros tantos no. Pero aparte de los problemas mencionados en esta Memoria también han surgido ideas que se pueden adaptar como futuras modificaciones de este TFG. A continuación se comentarán algunas de estas posibles ampliaciones.

Como se discutió cuando se hablaba de los espacios de colores se podrían probar nuevos espacios de colores para detectar las señales, podrían implementarse las ecuaciones necesarias para trabajar en CBM [5] y comparar los resultados de trabajar con ambos espacios.

Para solucionar los problemas de las señales a contraluz (las sombras producidas por objetos como árboles y edificios no se podrían corregir de esta manera) se podría probar un método por el cual mediante una IMU se sabría la orientación del vehículo y comparándola con la hora del día se podría tener una idea de que nivel de rojo van a tener las señales y así poder variar los umbrales de detección para adaptarnos a las condiciones lumínicas de cada momento. Esto permitiría disminuir los falsos positivos en buenas condiciones y aumentar la cantidad de verdaderos positivos en condiciones desfavorables, no obstante en condiciones de mala iluminación severa como hemos visto en Figura 21 y Figura 22 hay condiciones donde sería imposible distinguirlas del fondo. También podrían analizarse la cantidad de luz que hay en el momento (por ejemplo con fotorreceptores) y adaptar los parámetros en caso de que este nublado.

Otra opción sería realizar la búsqueda de las señales circulares mediante un clasificador tipo haar, del mismo tipo del usado para el reconocimiento. Este clasificador estaría entrenado con señales circulares de todo tipo y se pasaría por la imagen completa. Una vez entrenado se probaría su eficacia y se cruzarían los datos de los resultados con los obtenidos al usar *HoughCircles* para comparar cual es mejor.

En este proyecto sólo se ha tratado la detección de señales de prohibición y precaución. Sin embargo hay otras señales que deberían incluirse en un detector de señales de tráfico, como son las señales de obligación, recomendación y fin de prohibición. Para ello habría que cambiar los umbrales de colores para que capten los tonos azules, aunque aquí habría que tener precaución porque el cielo es azul y podría complicar la detección. También podría desarrollarse un sistema para detectar los carteles donde están las direcciones, pero sería más complicado porque habría que desarrollar también un OCR que las interpretara.

Como se ha tratado en el apartado de reconocimiento de datos, la base de imágenes positivas ha sido desarrollada por completo dentro del proyecto para tratar de optimizar los resultados, por tanto el número de muestras por señal no es muy alto, y en algunas señales, extremadamente bajo. En futuros trabajos podrían ampliarse la cantidad de señales de tráfico a

reconocer aumentando la base de datos de señales disponible. También se podrían variar los métodos de reconocimiento, como sería usar características tipo HOG dentro del clasificador de cascada, o pasar a usar otro método como una red neuronal o una SVM.

Una vez desarrollado el programa habría que implementarlo dentro del vehículo, para ello habría que mostrar al conductor por una interfaz la señal reconocida y avisarle si no está cumpliendo con lo que esta indica, en el caso de las señales de velocidad si está por encima de ella, o si se introduce en una vía de sentido contrario emitir una señal de aviso.

12. Conclusiones

La realización de este trabajo ha supuesto un reto al que enfrentarse y muchas horas de frustración delante de la pantalla buscando porque el programa no compilaba. Por otra parte me ha servido para adquirir muchos conocimientos sobre ROS, visión por ordenador, sistemas inteligentes en los vehículos.... y también y muy importante como se lleva a cabo un proyecto, las distintas fases que se van siguiendo (planificación, desarrollo, pruebas...) y la forma en la que se trabaja en él.

La realización de este trabajo de fin de grado ha sido para mí un reto que me ha servido para adquirir muchos conocimientos sobre sobre ROS, visión por ordenador y sistemas inteligentes en los vehículos. Me ha supuesto también muchas horas de frustración frente la pantalla buscando soluciones cuando el programa no compilaba. Pero he aprendido como se lleva a cabo un proyecto, las distintas fases de su evolución (planificación, desarrollo, pruebas...) y la forma en la que se trabaja en él, y todo esto es muy importante.

Los resultados finales del proyecto son bastante satisfactorios. Un 80% en la detección de señales con muy bajo número de falsos positivos (1,5%) y en torno al 88% en el reconocimiento, con un 5% de falsos positivos. Teniendo en cuenta las limitaciones comentadas durante el proyecto son unos resultados muy buenos.

El coste total del proyecto ha sido de 5.550 €, y la mayor parte de éste lo ha sido en personal (casi 5.300 €) por lo que realmente el coste del añadir este sistema a un automóvil resultaría sería muy bajo, más teniendo en cuenta que parte de ese coste (la cámara y el ordenador) podría ser compartido con otros sistemas del coche, lo que hace de este Proyecto de detección y reconocimiento de señales un sistema barato y eficiente.

Es decir, con un coste muy bajo se podría instalar en cualquier vehículo un sistema de detección y reconocimiento de señales de tráfico con una alta tasa de éxito. Esta tecnología permitiría no sólo evitar multas de tráfico sino disminuir el número de accidentes de tráfico, ya que como se vio en Problemática actual un gran número de accidentes se producen por superar la velocidad máxima permitida.

13. Bibliografía

- [1] OMS, «Informe mundial sobre la prevención de los traumatismos causados por el tránsito».
- [2] DGT, «Principales cifras de la siniestralidad vial en España 2013».
- [3] Ibáñez, «Xataka,» [En línea]. Available: <http://www.xataka.com/automovil/sistemas-de-deteccion-en-los-coches-para-evitar-accidentes>. [Último acceso: 13 8 2015].
- [4] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Hill_Descent_Control_system. [Último acceso: 22 9 2015].
- [5] Q. Z. a. S.-i. Kamata, «Improved Color Barycenter Model for Road-Sign Detection».
- [6] L. E. M. M. A. S. J. M. A. Arturo de la Escalera, «Road Traffic Sign Detection and Classification,» *IEEE*, 1997.
- [7] SJCAM, «SJ4000 WiFi 1080P Full HD Action Camera Sport DVR,» [En línea]. Available: http://www.sjcam.com/cameras/27-sjcam-sj4000-wifi-1080p-full-hd-action-camera-sport-dvr.html?search_query=4000+wifi&results=13#/color-white. [Último acceso: 1 2015].
- [8] UC3M, «IVVI 2.0: An intelligent vehicle based on computational perception».
- [9] OpenCV, «Camera Calibration and 3D Reconstruction».
- [10] ROS, «Camera_calibration Package Summary».
- [11] OpenCV, «Haar Feature-based Cascade Classifier for Object Detection».
- [12] M. M. T. T. B. M. Andreas Møgelmoose, «Learning to Detect Traffic Signs: Comparative Evaluation of Synthetic and Real World Data Sets».
- [13] N. Seo, «Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features),» [En línea]. Available: <http://note.sonots.com/SciSoftware/haartraining.html>. [Último acceso: 7 2015].
- [14] OpenCV, «Cascade Classifier Training».
- [15] M. Hacienda, «BOE - Tablas de amortización,» 2015. [En línea]. Available: <http://www.boe.es/buscar/act.php?id=BOE-A-2004-14600>.

